Online Low-Rank Subspace Clustering by Basis Dictionary Pursuit

Jie Shen
Dept. of Computer Science
Rutgers University
Piscataway, NJ 08854, USA
js2007@rutgers.edu

Ping Li
Dept. of Statistics & Biostatistics
Department of Computer Science
Rutgers University
Piscataway, NJ 08854, USA
pingli@stat.rutgers.edu

Huan Xu
Dept. of Industrial & Sys. Engineering
National University of Singapore
Singapore 117576, Singapore
isexuh@nus.edu.sg

Abstract

Low-Rank Representation (LRR) has been a significant method for segmenting data that are generated from a union of subspaces. It is also known that solving LRR is challenging in terms of time complexity and memory footprint, in that the size of the nuclear norm regularized matrix is n-by-n (where n is the number of samples). In this paper, we thereby develop a novel online implementation of LRR that reduces the memory cost from $\mathcal{O}(n^2)$ to $\mathcal{O}(pd)$, with p being the ambient dimension and d being some estimated rank ($d < p \ll n$). We also establish the theoretical guarantee that the sequence of solutions produced by our algorithm converges to a stationary point of the expected loss function asymptotically. Extensive experiments on synthetic and realistic datasets further substantiate that our algorithm is fast, robust and memory efficient.

Keywords: Subspace Clustering, Online Optimization, Low-Rank Matrix, Asymptotic Convergence

1 Introduction

In the past a few years, subspace clustering [Vid10, SC12] has been extensively studied and has established solid applications, for example, in computer vision [EV09] and network topology inference [EBN11]. Among many subspace clustering algorithms which aim to obtain a structured representation to fit the underlying data, two prominent examples are Sparse Subspace Clustering (SSC) [EV09, SEC14] and Low-Rank Representation (LRR) [LLY+13]. Both of them utilize the idea of self-expressiveness, i.e., expressing each sample as a linear combination of the remaining. What is of difference is that SSC pursues a sparse solution while LRR prefers a low-rank structure.

In this paper, we are interested in the LRR method, which is shown to achieve state-of-the-art performance on a broad range of real-world problems [LLY+13]. Recently, [LL14] demonstrated that, when equipped with a proper dictionary, LRR can even handle the coherent data – a challenging issue in the literature [CR09, CLMW11] but commonly emerges in realistic datasets such as the Netflix.

Formally, the LRR problem we investigate here is formulated as follows [LLY+13]:

$$\min_{\boldsymbol{X},\boldsymbol{E}} \frac{\lambda_1}{2} \left\| \boldsymbol{Z} - \boldsymbol{Y}\boldsymbol{X} - \boldsymbol{E} \right\|_F^2 + \left\| \boldsymbol{X} \right\|_* + \lambda_2 \left\| \boldsymbol{E} \right\|_1 . \tag{1.1}$$

Here, $\boldsymbol{Z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n) \in \mathbb{R}^{p \times n}$ is the observation matrix with n samples lying in a p-dimensional subspace. The matrix $\boldsymbol{Y} \in \mathbb{R}^{p \times n}$ is a given dictionary, $\boldsymbol{E}$ is some possible sparse corruption and λ_1 and λ_2 are two tunable parameters. Typically, $\boldsymbol{Y}$ is chosen as the dataset $\boldsymbol{Z}$ itself. The program seeks a low-rank

representation $\boldsymbol{X} \in \mathbb{R}^{n \times n}$ among all samples, each of which can be approximated by a linear combination of the atoms in the dictionary $\boldsymbol{Y}$.

While LLR is mathematically elegant, three issues are immediately incurred in the face of big data:

Issue 1 (Memory cost of $\boldsymbol{X}$)**.** In the LRR formulation (1.1), there is typically no sparsity assumption on $\boldsymbol{X}$. Hence, the memory footprint of $\boldsymbol{X}$ is proportional to n^2 which precludes most of the recently developed nuclear norm solvers [LCM10, JS10, AKKS12, HO14].

Issue 2 (Computational cost of $\|\boldsymbol{X}\|_*$)**.** Due to the size of the nuclear norm regularized matrix $\boldsymbol{X}$ is $n \times n$, optimizing such problems can be computationally expensive even when n is not too large [RFP10].

Issue 3 (Memory cost of $\boldsymbol{Y}$)**.** Since the dictionary size is $p \times n$, it is prohibitive to store the entire dictionary $\boldsymbol{Y}$ during optimization when manipulating a huge volume of data.

To remedy these issues, especially the memory bottleneck, one potential way is solving the problem in online manner. That is, we sequentially reveal the samples $\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n$ and update the components in $\boldsymbol{X}$ and $\boldsymbol{E}$. Nevertheless, such strategy appears difficult to execute due the the residual term in (1.1). To be more precise, we note that each column of $\boldsymbol{X}$ is the coefficients of a sample with respect to the *entire* dictionary $\boldsymbol{Y}$, e.g., $\boldsymbol{z}_1 \approx \boldsymbol{Y}\boldsymbol{x}_1 + \boldsymbol{e}_1$. This indicates that without further technique, we have to load the entire dictionary Y so as to update the columns of $\boldsymbol{X}$. Hence, for our purpose, we need to tackle a more serious challenge:

Issue 4 (Partial realization of $\boldsymbol{Y}$)**.** We are required to guarantee the optimality of the solution but can only access part of the atoms of Y in each iteration.

1.1 Related Works

There are a vast body of works attempting to mitigate the memory and computational bottleneck of the nuclear norm regularizer. However, to the best of our knowledge, none of them can handle Issue 3 and Issue 4 in the LRR problem.

One of the most popular ways to alleviate the huge memory cost is online implementation. [FXY13] devised an online algorithm for the Robust Principal Component Analysis (RPCA) problem, which makes the memory cost independent of the sample size. Yet, compared to RPCA where the size of the nuclear norm regularized matrix is $p \times n$, that of LRR is $n \times n$ – a worse and more challenging case. Moreover, their algorithm cannot address the partial dictionary issue that emerges in our case. It is also worth mentioning that [QVLH14] established another online variant of RPCA. But since we are dealing with a different problem setting, i.e., the multiple subspaces regime, it is not clear how to extend their method to LRR.

To tackle the computational overhead, [CCS10] considered singular value thresholding technique. However, it is not scalable to large problems since it calls singular value decomposition (SVD) in each iteration. [JS10] utilized a sparse semi-definite programming solver to derive a simple yet efficient algorithm. Unfortunately, the memory requirement of their algorithm is proportional to the number of observed entries, making it impractical when the regularized matrix is large and dense (which is the case of LRR). [AKKS12] combined stochastic subgradient and incremental SVD to boost efficiency. But for the LRR problem, the type of the loss function does not meet the requirements and thus, it is still not practical to use that algorithm in our case.

Another line in the literature explores a structured formulation of LRR beyond the low-rankness. For example, [WXL13] provably showed that combining LRR with SSC can take advantages of both methods. [LL14] demonstrated that LRR is able to cope with the intrinsic group structure of the data. Very recently, [SL16] argued that the vanilla LRR program does not fully characterize the nature of multiple subspaces, and presented several effective alternatives to LRR.

1.2 Summary of Contributions

In this paper, we propose a new algorithm called Online Low-Rank Subspace Clustering (OLRSC), which admits a low computational complexity. In contrast to existing solvers, OLRSC reduces the memory cost of LRR from $\mathcal{O}(n^2)$ to $\mathcal{O}(pd)$ ($d < p \ll n$). This nice property makes OLRSC an appealing solution for large-scale subspace clustering problems. Furthermore, we prove that the sequence of solutions produced by OLRSC converges to a stationary point of the expected loss function asymptotically even though only one atom of $\boldsymbol{Y}$ is available at each iteration. In a nutshell, OLRSC resolves *all* practical issues of LRR and still promotes global low-rank structure – the merit of LRR.

1.3 Roadmap

The paper is organized as follows. In Section 2, we reformulate the LRR program (1.1) in a way which is amenable for online optimization. Section 3 presents the algorithm that incrementally minimizes a surrogate function to the empirical loss. Along with that, we establish a theoretical guarantee in Section 4. The experimental study in Section 5 confirms the efficacy and efficiency of our proposed algorithm. Finally, we conclude the work in Section 6 and the lengthy proof is deferred to the appendix.

Notation. We use bold lowercase letters, e.g. $\boldsymbol{v}$, to denote a column vector. The ℓ_2 norm and ℓ_1 norm of a vector $\boldsymbol{v}$ are denoted by $\|\boldsymbol{v}\|_2$ and $\|\boldsymbol{v}\|_1$ respectively. Bold capital letters such as $\boldsymbol{M}$ are used to denote a matrix, and its transpose is denoted by $\boldsymbol{M}^\top$. For an invertible matrix $\boldsymbol{M}$, we write its inverse as $\boldsymbol{M}^{-1}$. The capital letter $\boldsymbol{I}_r$ is reserved for identity matrix where r indicates the size. The jth column of a matrix $\boldsymbol{M}$ is denoted by $\boldsymbol{m}_j$ if not specified. Three matrix norms will be used: $\|\boldsymbol{M}\|_*$ for the nuclear norm, i.e., the sum of the singular values, $\|\boldsymbol{M}\|_F$ for the Frobenius norm and $\|\boldsymbol{M}\|_1$ for the ℓ_1 norm of a matrix seen as a long vector. The trace of a square matrix $\boldsymbol{M}$ is denoted by $\mathrm{Tr}(\boldsymbol{M})$. For an integer $n > 0$, we use $[n]$ to denote the integer set $\{1, 2, \cdots, n\}$.

2 Problem Formulation

Our goal is to efficiently learn the representation matrix $\boldsymbol{X}$ and the corruption matrix $\boldsymbol{E}$ in an online manner so as to mitigate the issues mentioned in Section 1. The first technique for our purpose is a *non-convex reformulation* of the nuclear norm. Assume that the rank of the global optima $\boldsymbol{X}$ in (1.1) is at most d. Then a standard result in the literature (see, e.g., [FHB01]) showed that,

$$\|\boldsymbol{X}\|_* = \min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{X}=\boldsymbol{U}\boldsymbol{V}^\top} \frac{1}{2}\left(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2\right), \tag{2.1}$$

where $\boldsymbol{U} \in \mathbb{R}^{n\times d}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times d}$. The minimum can be attained at, for example, $\boldsymbol{U} = \boldsymbol{U}_0\boldsymbol{S}_0^{\frac{1}{2}}$ and $\boldsymbol{V} = \boldsymbol{V}_0\boldsymbol{S}_0^{\frac{1}{2}}$ where $\boldsymbol{X} = \boldsymbol{U}_0\boldsymbol{S}_0\boldsymbol{V}_0^\top$ is the singular value decomposition.

In this way, (1.1) can be written as follows:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}} \frac{\lambda_1}{2}\left\|\boldsymbol{Z} - \boldsymbol{Y}\boldsymbol{U}\boldsymbol{V}^\top - \boldsymbol{E}\right\|_F^2 + \frac{1}{2}\|\boldsymbol{U}\|_F^2 + \frac{1}{2}\|\boldsymbol{V}\|_F^2 + \lambda_2\|\boldsymbol{E}\|_1. \tag{2.2}$$

Note that by this reformulation, updating the entries in $\boldsymbol{X}$ amounts to sequentially updating the rows of $\boldsymbol{U}$ and $\boldsymbol{V}$. Also note that this technique is utilized in [FXY13] for online RPCA. Unfortunately, the size of $\boldsymbol{U}$ and $\boldsymbol{V}$ in our problem are both proportional to n and the dictionary $\boldsymbol{Y}$ is partially observed in each iteration, making the algorithm in [FXY13] not applicable to LRR. Related to online implementation, another challenge is that, all the rows of $\boldsymbol{U}$ are coupled together at this moment as $\boldsymbol{U}$ is left multiplied by $\boldsymbol{Y}$ in the first term. This makes it difficult to sequentially update the rows of $\boldsymbol{U}$.

For the sake of decoupling the rows of $\boldsymbol{U}$, as part of the crux of our technique, we introduce an auxiliary variable $\boldsymbol{D} = \boldsymbol{Y}\boldsymbol{U}$, whose size is $p \times d$ (i.e., independent of the sample size n). Interestingly, in this way, we are approximating the term $\boldsymbol{Z} - \boldsymbol{E}$ with $\boldsymbol{D}\boldsymbol{V}^\top$, which provides an intuition on the role of $\boldsymbol{D}$: Namely, $\boldsymbol{D}$ can be seen as a *basis dictionary* of the clean data, with $\boldsymbol{V}$ being the coefficients.

These key observations allow us to derive an equivalent reformulation of LRR (1.1):

$$\min_{\boldsymbol{D},\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}} \frac{\lambda_1}{2} \left\| \boldsymbol{Z} - \boldsymbol{Y}\boldsymbol{U}\boldsymbol{V}^\top - \boldsymbol{E} \right\|_F^2 + \frac{1}{2}\left(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2 \right) + \lambda_2 \|\boldsymbol{E}\|_1, \quad \text{s.t. } \boldsymbol{D} = \boldsymbol{Y}\boldsymbol{U}. \tag{2.3}$$

By penalizing the constraint in the objective, we obtain a *regularized* version of LRR on which our algorithm is based:

$$\min_{\boldsymbol{D},\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}} \frac{\lambda_1}{2} \left\| \boldsymbol{Z} - \boldsymbol{D}\boldsymbol{V}^\top - \boldsymbol{E} \right\|_F^2 + \frac{1}{2}\left(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2 \right) + \lambda_2 \|\boldsymbol{E}\|_1 + \frac{\lambda_3}{2} \|\boldsymbol{D} - \boldsymbol{Y}\boldsymbol{U}\|_F^2. \tag{2.4}$$

Remark 1 (Superiority to LRR)**.** There are two advantages of (2.4) compared to (1.1). First, it is amenable for online optimization. Second, it is more informative since it explicitly models the basis of the union of subspaces, hence a better subspace recovery and clustering (see Section 5). This actually meets the core idea of [LL14] but they *assumed* Y contains true subspaces whereas we *learn* the true subspaces.

Remark 2 (Parameter)**.** Note that λ_3 may be gradually increased until some maximum value is attained so as to enforce the equality constraint. In this way, (2.4) attains the same minimum as (1.1). Actually, the choice of λ_3 depends on how much information $\boldsymbol{Y}$ brings for the subspace basis. As we aforementioned, $\boldsymbol{D}$ is the basis dictionary of the clean data and is in turn approximated by (or equal to) $\boldsymbol{Y}\boldsymbol{U}$. This suggests that the range of $\boldsymbol{D}$ is a subset of that of $\boldsymbol{Y}$. As a typical choice of $\boldsymbol{Y} = \boldsymbol{Z}$, if $\boldsymbol{Z}$ is slightly corrupted, we would like to pick a large quantity for λ_3.

Remark 3 (Connection to RPCA)**.** Due to our explicit modeling of the basis, we unify LRR and RPCA as follows: for LRR, $\boldsymbol{D} \approx \boldsymbol{Y}\boldsymbol{U}$ (or $\boldsymbol{D} = \boldsymbol{Y}\boldsymbol{U}$ if λ_3 tends to infinity) while for RPCA, $\boldsymbol{D} = \boldsymbol{U}$. That is, ORPCA [FXY13] considers a problem of $\boldsymbol{Y} = \boldsymbol{I}_p$ whose size is independent of n, hence can be kept in memory which naturally resolves Issue 3 and 4. This is why RPCA can be easily implemented in an online fashion while LRR cannot.

Remark 4 (Connection to Dictionary Learning)**.** Generally speaking, LRR (1.1) can be seen as a coding algorithm, with the dictionary $\boldsymbol{Y}$ known in advance and $\boldsymbol{X}$ is a desired structured code while other popular algorithms such as dictionary learning (DL) [MBPS10] simultaneously optimizes the dictionary and the sparse code. Interestingly, in view of (2.4), the link of LRR and DL becomes more clear in the sense that the difference lies in the way how the dictionary is constrained. That is, for LRR we have $\boldsymbol{D} \approx \boldsymbol{Y}\boldsymbol{U}$ and $\boldsymbol{U}$ is further regularized by Frobenius norm whereas for DL, we have $\|\boldsymbol{d}_i\|_2 \leq 1$ for each column of D.

Let $\boldsymbol{z}_i$, $\boldsymbol{y}_i$, $\boldsymbol{e}_i$, $\boldsymbol{u}_i$, and $\boldsymbol{v}_i$ be the ith column of matrices $\boldsymbol{Z}$, $\boldsymbol{Y}$, $\boldsymbol{E}$, $\boldsymbol{U}^\top$ and $\boldsymbol{V}^\top$ respectively and define

$$\tilde{\ell}(\boldsymbol{z}, \boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e}) \overset{\text{def}}{=} \frac{\lambda_1}{2} \|\boldsymbol{z} - \boldsymbol{D}\boldsymbol{v} - \boldsymbol{e}\|_2^2 + \frac{1}{2}\|\boldsymbol{v}\|_2^2 + \lambda_2 \|\boldsymbol{e}\|_1, \tag{2.5}$$

$$\ell(\boldsymbol{z}, \boldsymbol{D}) = \min_{\boldsymbol{v},\boldsymbol{e}} \tilde{\ell}(\boldsymbol{z}, \boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e}). \tag{2.6}$$

In addition, let

$$\tilde{h}(\boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{U}) \overset{\text{def}}{=} \sum_{i=1}^{n} \frac{1}{2}\|\boldsymbol{u}_i\|_2^2 + \frac{\lambda_3}{2} \left\| \boldsymbol{D} - \sum_{i=1}^{n} \boldsymbol{y}_i \boldsymbol{u}_i^\top \right\|_F^2, \tag{2.7}$$

$$h(\boldsymbol{Y}, \boldsymbol{D}) = \min_{\boldsymbol{U}} \tilde{h}(\boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{U}). \tag{2.8}$$

Then (2.4) can be rewritten as:

$$\min_{\boldsymbol{D}} \min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}} \sum_{i=1}^{n} \tilde{\ell}(\boldsymbol{z}_i, \boldsymbol{D}, \boldsymbol{v}_i, \boldsymbol{e}_i) + \tilde{h}(\boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{U}), \tag{2.9}$$

which amounts to minimizing the empirical loss function:

$$f_n(\boldsymbol{D}) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \ell(\boldsymbol{z}_i, \boldsymbol{D}) + \frac{1}{n} h(\boldsymbol{Y}, \boldsymbol{D}). \tag{2.10}$$

In stochastic optimization, we are interested in analyzing the optimality of the obtained solution with respect to the expected loss function. To this end, we first derive the optimal solutions $\boldsymbol{U}^*$, $\boldsymbol{V}^*$ and $\boldsymbol{E}^*$ that minimize (2.9) which renders a concrete form of the empirical loss function $f_n(\boldsymbol{D})$, hence we are able to derive the expected loss.

Given $\boldsymbol{D}$, we need to compute the optimal solutions $\boldsymbol{U}^*$, $\boldsymbol{V}^*$ and $\boldsymbol{E}^*$ to evaluate the objective value of $f_n(\boldsymbol{D})$. What is of interest here is that, the optimization procedure of $\boldsymbol{U}$ is totally different from that of $\boldsymbol{V}$ and $\boldsymbol{E}$. According to (2.6), when $\boldsymbol{D}$ is given, each $\boldsymbol{v}_i^*$ and $\boldsymbol{e}_i^*$ can be solved by only accessing the ith sample $\boldsymbol{z}_i$. However, the optimal $\boldsymbol{u}_i^*$ depends on the whole dictionary $\boldsymbol{Y}$ as the second term in $\tilde{h}(\boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{U})$ couples all the $\boldsymbol{u}_i$'s. Fortunately, it is possible to obtain a closed form solution for $\boldsymbol{U}^*$ which simplifies our analysis. To be more precise, the first order optimality condition for (2.8) gives

$$\frac{\partial \tilde{h}(\boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{U})}{\partial \boldsymbol{U}} = \boldsymbol{U} + \lambda_3 (\boldsymbol{Y}^\top \boldsymbol{Y} \boldsymbol{U} - \boldsymbol{Y}^\top \boldsymbol{D}) = 0, \tag{2.11}$$

which implies

$$\begin{aligned}
\boldsymbol{U}^* &= \left(\lambda_3{}^{-1} \boldsymbol{I}_p + \boldsymbol{Y}^\top \boldsymbol{Y}\right)^{-1} \boldsymbol{Y}^\top \boldsymbol{D} \\
&= \lambda_3 \sum_{j=0}^{+\infty} \left(-\lambda_3 \boldsymbol{Y}^\top \boldsymbol{Y}\right)^j \boldsymbol{Y}^\top \boldsymbol{D} \\
&= \lambda_3 \boldsymbol{Y}^\top \left[\sum_{j=0}^{+\infty} \left(-\lambda_3 \boldsymbol{Y} \boldsymbol{Y}^\top\right)^j \right] \boldsymbol{D} \\
&= \boldsymbol{Y}^\top \left(\lambda_3{}^{-1} \boldsymbol{I}_p + \boldsymbol{Y} \boldsymbol{Y}^\top\right)^{-1} \boldsymbol{D}.
\end{aligned} \tag{2.12}$$

Likewise, another component $\boldsymbol{Y}\boldsymbol{U}^{*\top}$ in (2.7) can be derived as follows:

$$\boldsymbol{Y}\boldsymbol{U}^{*\top} = \boldsymbol{D} - \frac{1}{n} \left(\frac{1}{n} \boldsymbol{I}_p + \frac{\lambda_3}{n} \boldsymbol{N}_n \right)^{-1} \boldsymbol{D}, \tag{2.13}$$

where we denote

$$\boldsymbol{N}_n = \sum_{i=1}^{n} \boldsymbol{y}_i \boldsymbol{y}_i^\top. \tag{2.14}$$

Recall that $\boldsymbol{u}_i$ is the ith column of $\boldsymbol{U}^\top$. So for each $i \in [n]$, we immediately have

$$\boldsymbol{u}_i^* = \boldsymbol{D}^\top \left(\frac{1}{\lambda_3} \boldsymbol{I}_p + \sum_{i=1}^{n} \boldsymbol{y}_i \boldsymbol{y}_i^\top \right)^{-1} \boldsymbol{y}_i = \frac{1}{n} \boldsymbol{D}^\top \left(\frac{1}{\lambda_3 n} \boldsymbol{I}_p + \frac{1}{n} \boldsymbol{N}_n \right)^{-1} \boldsymbol{y}_i. \tag{2.15}$$

Plugging $\boldsymbol{U}^*$ and $\boldsymbol{Y}\boldsymbol{U}^{*\top}$ back to $\tilde{h}(\boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{U})$ gives

$$h(\boldsymbol{Y}, \boldsymbol{D}) = \frac{1}{n^2}\sum_{i=1}^{n}\frac{1}{2}\left\|\boldsymbol{D}^\top\left(\frac{1}{\lambda_3 n}\boldsymbol{I}_p + \frac{1}{n}\boldsymbol{N}_n\right)^{-1}\boldsymbol{y}_i\right\|_2^2 + \frac{\lambda_3}{2n^2}\left\|\left(\frac{1}{n}\boldsymbol{I}_p + \frac{\lambda_3}{n}\boldsymbol{N}_n\right)^{-1}\boldsymbol{D}\right\|_F^2. \tag{2.16}$$

Now we derive the expected loss function, which is defined as the limit of the empirical loss function when n tends to infinity. If we assume that all the samples are drawn independently and identically from some (unknown) distribution, we have

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\ell(\boldsymbol{z}_i, \boldsymbol{D}) = \mathbb{E}_{\boldsymbol{z}}[\ell(\boldsymbol{z}, \boldsymbol{D})]. \tag{2.17}$$

If we further assume that the smallest singular value of $\frac{1}{n}\boldsymbol{N}_n$ is bounded away from zero (which implies $\boldsymbol{N}_n$ is invertible and the spectrum of $\boldsymbol{N}_n^{-1}$ is bounded from the above), we have

$$0 \leq \lim_{n\to\infty}\frac{1}{n}h(\boldsymbol{Y}, \boldsymbol{D}) \leq \lim_{n\to\infty}\frac{1}{n^3}\sum_{i=1}^{n}\mathrm{C}_0 = 0. \tag{2.18}$$

Here C_0 is some absolute constant since $\boldsymbol{D}$ is fixed and $\boldsymbol{y}_i$'s are bounded. Hence, it follows that

$$\lim_{n\to\infty}\frac{1}{n}h(\boldsymbol{Y}, \boldsymbol{D}) = 0. \tag{2.19}$$

Finally, the expected loss function is given by

$$f(\boldsymbol{D}) \stackrel{\text{def}}{=} \lim_{n\to\infty} f_n(\boldsymbol{D}) = \mathbb{E}_{\boldsymbol{z}}\big[\ell(\boldsymbol{z}, \boldsymbol{D})\big]. \tag{2.20}$$

3 Algorithm

Our OLRSC algorithm is summarized in Algorithm 1. Recall that OLRSC is an online implementation to solve (2.10), which is derived from the regularized version of LRR (2.4). The main idea is optimizing the variables in an alternative manner. That is, at the t-th iteration, assume the basis dictionary $\boldsymbol{D}_{t-1}$ is given, we compute the optimal solutions $\{\boldsymbol{v}_t, \boldsymbol{e}_t\}$ by minimizing the objective function $\tilde{\ell}(\boldsymbol{z}_t, \boldsymbol{D}_{t-1}, \boldsymbol{v}, \boldsymbol{e})$ over $\boldsymbol{v}$ and $\boldsymbol{e}$. For $\boldsymbol{u}_t$, we need a more carefully designed paradigm since a direct optimization involves loading the full dictionary $\boldsymbol{Y}$ (see (2.15)). We will elaborate the details later. Subsequently, we update the basis dictionary $\boldsymbol{D}_t$ by optimizing a surrogate function to the empirical loss $f_n(\boldsymbol{D})$. In our algorithm, we need to maintain three additional accumulation matrices for which the sizes are independent of n.

Solving $\{\boldsymbol{v}_t, \boldsymbol{e}_t\}$. We observe that if $\boldsymbol{e}$ is fixed, we can optimize $\boldsymbol{v}$ in closed form:

$$\boldsymbol{v} = \left(\lambda_1^{-1}\boldsymbol{I}_d + \boldsymbol{D}_{t-1}^\top\boldsymbol{D}_{t-1}\right)^{-1}\boldsymbol{D}_{t-1}^\top(\boldsymbol{z}_t - \boldsymbol{e}). \tag{3.1}$$

Conversely, given $\boldsymbol{v}$, the variable $\boldsymbol{e}$ is obtained via soft-thresholding [Don95]:

$$\boldsymbol{e} = \mathcal{S}_{\lambda_2/\lambda_1}(\boldsymbol{z}_t - \boldsymbol{D}_{t-1}\boldsymbol{v}). \tag{3.2}$$

Thus, we utilize block coordinate minimization algorithm to optimize $\boldsymbol{v}$ and $\boldsymbol{e}$.

Algorithm 1 Online Low-Rank Subspace Clustering

Require: $\boldsymbol{Z} \in \mathbb{R}^{p \times n}$ (observed samples), $\boldsymbol{Y} \in \mathbb{R}^{p \times n}$, parameters λ_1, λ_2 and λ_3, initial basis $\boldsymbol{D}_0 \in \mathbb{R}^{p \times d}$, zero matrices $\boldsymbol{M}_0 \in \mathbb{R}^{p \times d}$, $\boldsymbol{A}_0 \in \mathbb{R}^{d \times d}$ and $\boldsymbol{B}_0 \in \mathbb{R}^{p \times d}$.

Ensure: Optimal basis $\boldsymbol{D}_n$.

for $t = 1$ to n **do**

Access the t-th sample $\boldsymbol{z}_t$ and the t-th atom $\boldsymbol{y}_t$.

Compute the coefficient and noise:

$$\{\boldsymbol{v}_t, \boldsymbol{e}_t\} = \arg\min_{\boldsymbol{v},\boldsymbol{e}} \tilde{\ell}(\boldsymbol{z}_t, \boldsymbol{D}_{t-1}, \boldsymbol{v}, \boldsymbol{e}),$$

$$\boldsymbol{u}_t = \arg\min_{\boldsymbol{u}} \tilde{\ell}_2(\boldsymbol{y}_t, \boldsymbol{D}_{t-1}, \boldsymbol{M}_{t-1}, \boldsymbol{u}).$$

Update the accumulation matrices:

$$\boldsymbol{M}_t \leftarrow \boldsymbol{M}_{t-1} + \boldsymbol{y}_t \boldsymbol{u}_t^\top, \;\; \boldsymbol{A}_t \leftarrow \boldsymbol{A}_{t-1} + \boldsymbol{v}_t \boldsymbol{v}_t^\top, \;\; \boldsymbol{B}_t \leftarrow \boldsymbol{B}_{t-1} + (\boldsymbol{z}_t - \boldsymbol{e}_t)\boldsymbol{v}_t^\top.$$

Update the basis dictionary:

$$\boldsymbol{D}_t = \arg\min_{\boldsymbol{D}} \frac{1}{t}\left[\frac{1}{2}\operatorname{Tr}\left(\boldsymbol{D}^\top \boldsymbol{D}(\lambda_1 \boldsymbol{A}_t + \lambda_3 \boldsymbol{I}_d)\right) - \operatorname{Tr}\left(\boldsymbol{D}^\top(\lambda_1 \boldsymbol{B}_t + \lambda_3 \boldsymbol{M}_t)\right)\right].$$

end for

Solving $\boldsymbol{u}_t$. The closed form solution (2.15) tells us that it is impossible to derive an accurate estimation of $\boldsymbol{u}_t$ without the entire dictionary $\boldsymbol{Y}$. Thus, we have to "approximately" solve it during the online optimization procedure[1].

Our carefully designed approximate process to solve $\tilde{h}(\boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{U})$ (2.7) is motivated by the coordinate minimization method appealing to $\tilde{h}(\boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{U})$. As a convention, such method starts with initial guess that $\boldsymbol{u}_i = \boldsymbol{0}$ for all $i \in [n]$ and updates the $\boldsymbol{u}_i$'s in a cyclic order, i.e., $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\cdots$, $\boldsymbol{u}_n$, $\boldsymbol{u}_1$, $\cdots$. Let us consider the first pass where we have already updated $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\cdots$, $\boldsymbol{u}_{t-1}$ and are to optimize over $\boldsymbol{u}_t$ for some $t > 0$. Note that since the initial values are zero, $\boldsymbol{u}_{t+1} = \boldsymbol{u}_{t+2} = \cdots = \boldsymbol{u}_n = \boldsymbol{0}$. Thereby, the optimal $\boldsymbol{u}_t$ is actually given by minimizing the following function:

$$\tilde{\ell}_2(\boldsymbol{y}_t, \boldsymbol{D}, \boldsymbol{M}_{t-1}, \boldsymbol{u}) \overset{\text{def}}{=} \frac{1}{2}\left\|\boldsymbol{u}\right\|_2^2 + \frac{\lambda_3}{2}\left\|\boldsymbol{D} - \boldsymbol{M}_{t-1} - \boldsymbol{y}_t \boldsymbol{u}^\top\right\|_F^2, \tag{3.3}$$

where

$$\boldsymbol{M}_{t-1} = \sum_{i=1}^{t-1} \boldsymbol{y}_i \boldsymbol{u}_i^\top. \tag{3.4}$$

We easily obtain the closed form solution to (3.3) as follows:

$$\boldsymbol{u}_t = (\|\boldsymbol{y}_t\|_2^2 + 1/\lambda_3)^{-1}(\boldsymbol{D} - \boldsymbol{M}_{t-1})^\top \boldsymbol{y}_t. \tag{3.5}$$

Now let us turn to the alternating minimization algorithm, where $\boldsymbol{D}$ is updated iteratively rather than fixed as in (3.5). The above coordinate minimization process can be adjusted in this scenario as we did in Algorithm 1. That is, given $\boldsymbol{D}_{t-1}$, after revealing a new atom $\boldsymbol{y}_t$, we compute $\boldsymbol{u}_t$ by minimizing

[1]Here, "accurately" and "approximately" mean that when only $\boldsymbol{D}_{t-1}$, $\boldsymbol{z}_t$ and $\boldsymbol{y}_t$ are given, whether we can obtain the same solution $\{\boldsymbol{v}_t, \boldsymbol{e}_t, \boldsymbol{u}_t\}$ as for the batch problem (2.10).

$\tilde{\ell}_2(\boldsymbol{y}_t, \boldsymbol{D}_{t-1}, \boldsymbol{M}_{t-1}, \boldsymbol{u})$, followed by updating $\boldsymbol{D}_t$. In this way, when the algorithm terminates, we in essence run a one-pass update on $\boldsymbol{u}_t$'s with a simultaneous computation of new basis dictionary.

Solving $\boldsymbol{D}_t$. As soon as the past filtration $\{\boldsymbol{v}_i, \boldsymbol{e}_i, \boldsymbol{u}_i\}_{i=1}^t$ are available, we can compute a new iterate $\boldsymbol{D}_t$ by optimizing the surrogate function

$$g_t(\boldsymbol{D}) \overset{\text{def}}{=} \frac{1}{t}\left(\sum_{i=1}^{t} \tilde{\ell}(\boldsymbol{z}_i, \boldsymbol{D}, \boldsymbol{v}_i, \boldsymbol{e}_i) + \sum_{i=1}^{t} \frac{1}{2}\left\|\boldsymbol{u}_i\right\|_2^2 + \frac{\lambda_3}{2}\left\|\boldsymbol{D} - \boldsymbol{M}_t\right\|_F^2\right). \tag{3.6}$$

Expanding the first term, we find that $\boldsymbol{D}_t$ is given by

$$\begin{aligned}\boldsymbol{D}_t &= \arg\min_{\boldsymbol{D}} \frac{1}{t}\left[\frac{1}{2}\operatorname{Tr}\left(\boldsymbol{D}^\top \boldsymbol{D}(\lambda_1 \boldsymbol{A}_t + \lambda_3 \boldsymbol{I}_d)\right) - \operatorname{Tr}\left(\boldsymbol{D}^\top(\lambda_1 \boldsymbol{B}_t + \lambda_3 \boldsymbol{M}_t)\right)\right] \\ &= (\lambda_1 \boldsymbol{B}_t + \lambda_3 \boldsymbol{M}_t)(\lambda_1 \boldsymbol{A}_t + \lambda_3 \boldsymbol{I}_d)^{-1},\end{aligned} \tag{3.7}$$

where $\boldsymbol{A}_t = \sum_{i=1}^t \boldsymbol{v}_i \boldsymbol{v}_i^\top$ and $\boldsymbol{B}_t = \sum_{i=1}^t (\boldsymbol{z}_i - \boldsymbol{e}_i)\boldsymbol{v}_i^\top$. We point out that the size of $\boldsymbol{A}_t$ is $d \times d$ and that of $\boldsymbol{B}_t$ is $p \times d$, i.e., independent of sample size. In practice, as [MBPS10] suggested, one may apply a block coordinate descent approach to minimize over $\boldsymbol{D}$. Compared to the closed form solution given above, such algorithm usually converges very fast after revealing sufficient number of samples. In fact, we observe that a one-pass update on the columns of $\boldsymbol{D}$ suffices to ensure a favorable performance. See Algorithm 2.

Algorithm 2 Solving $\boldsymbol{D}$

Require: $\boldsymbol{D} \in \mathbb{R}^{p \times d}$ in the previous iteration, accumulation matrix $\boldsymbol{M}$, $\boldsymbol{A}$ and $\boldsymbol{B}$, parameters λ_1 and λ_3.
Ensure: Optimal $\boldsymbol{D}$ (updated).

1: Denote $\widehat{\boldsymbol{A}} = \lambda_1 \boldsymbol{A} + \lambda_3 \boldsymbol{I}$ and $\widehat{\boldsymbol{B}} = \lambda_1 \boldsymbol{B} + \lambda_3 \boldsymbol{M}$.
2: **repeat**
3:   **for** $j = 1$ to d **do**
4:     Update the jth column of $\boldsymbol{D}$:

$$\boldsymbol{d}_j \leftarrow \boldsymbol{d}_j - \frac{1}{\widehat{A}_{jj}}\left(\boldsymbol{D}\widehat{\boldsymbol{a}}_j - \widehat{\boldsymbol{b}}_j\right)$$

5:   **end for**
6: **until** convergence

Memory Cost. It is remarkable that the memory cost of Algorithm 1 is $\mathcal{O}(pd)$. To see this, note that when solving $\boldsymbol{v}_t$ and $\boldsymbol{e}_t$, we load the auxiliary variable $\boldsymbol{D}_t$ and a sample $\boldsymbol{z}_t$ into the memory, which costs $\mathcal{O}(pd)$. To compute the optimal $\boldsymbol{u}_t$'s, we need to access $\boldsymbol{D}_t$ and $\boldsymbol{M}_t \in \mathbb{R}^{p\times d}$. Although we aim to minimize (3.6), which seems to require all the past information, we actually only need to record $\boldsymbol{A}_t$, $\boldsymbol{B}_t$ and $\boldsymbol{M}_t$, whose sizes are at most $\mathcal{O}(pd)$ (since $d < p$).

Computational Efficiency. In addition to memory efficiency, we further clarify that the the computation in each iteration is cheap. To compute $\{\boldsymbol{v}_t, \boldsymbol{e}_t\}$, one may utilize the block coordinate method in [RT14] which enjoys linear convergence due to strong convexity. One may also apply the stochastic variance reduced algorithms which also ensure a geometric rate of convergence [XZ14, DBL14]. The $\boldsymbol{u}_t$ is given by simple matrix-vector multiplication, which costs $\mathcal{O}(pd)$. It is easy to see the update on the accumulation matrices is $\mathcal{O}(pd)$ and that of $\boldsymbol{D}_t$ is $\mathcal{O}(pd^2)$.

A Fully Online Scheme. Now we have provided a way to (approximately) optimize the LRR problem (1.1) in online fashion. Usually, researchers in the literature will take an optional post-processing step to refine

the segmentation accuracy, for example, applying spectral clustering on the representation matrix $\boldsymbol{X}$. In this case, one has to collect all the $\boldsymbol{u}_i$'s and $\boldsymbol{v}_i$'s to compute $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\top$ which again increases the memory cost to $\mathcal{O}(n^2)$. Here, we suggest an alternative scheme which admits $\mathcal{O}(kd)$ memory usage where k is the number of subspaces. The idea is utilizing the well-known k-means on $\boldsymbol{v}_i$'s. There are two notable advantages compared to the spectral clustering. First, updating the k-means model can be implemented in online manner and the computation is $\mathcal{O}(kd)$ for each iteration. Second, we observe that $\boldsymbol{v}_i$ is actually a robust feature for the ith sample. Combining the online k-means with Algorithm 1, we obtain a fully and efficient online subspace clustering scheme where the memory cost is $\mathcal{O}(pd)$. For the reader's convenience, we summarize this pipeline in Algorithm 3.

Algorithm 3 Fully Online Pipeline for Low-Rank Subspace Clustering

---

Require: $\boldsymbol{Z} \in \mathbb{R}^{p\times n}$ (observed samples), $\boldsymbol{Y} \in \mathbb{R}^{p\times n}$, parameters λ_1, λ_2 and λ_3, initial basis $\boldsymbol{D}_0 \in \mathbb{R}^{p\times d}$, zero matrices $\boldsymbol{M}_0 \in \mathbb{R}^{p\times d}$, $\boldsymbol{A}_0 \in \mathbb{R}^{d\times d}$ and $\boldsymbol{B}_0 \in \mathbb{R}^{p\times d}$, number of clusters k, initial centroids $\boldsymbol{C} \in \mathbb{R}^{d\times k}$.

Ensure: Optimal basis $\boldsymbol{D}_n$, cluster centroids $\boldsymbol{C}$, cluster assignments $\{o_1, o_2, \cdots, o_n\}$.

1: Initialize $r_1 = r_2 = \cdots = r_k = 0$.
2: **for** $t = 1$ to n **do**
3: Access the t-th sample $\boldsymbol{z}_t$ and the t-th atom $\boldsymbol{y}_t$.
4: Compute $\{\boldsymbol{v}_t, \boldsymbol{e}_t, \boldsymbol{u}_t, \boldsymbol{D}_t\}$ by Algorithm 1.
5: Compute $o_t = \arg\min_{1\leq j\leq k} \|\boldsymbol{v}_t - \boldsymbol{c}_j\|_2$.
6: Update the o_t-th center:

$$\begin{aligned} r_{o_t} &\leftarrow r_{o_t} + 1, \\ \boldsymbol{c}_{o_t} &\leftarrow \frac{r_{o_t} - 1}{r_{o_t}} \boldsymbol{c}_{o_t} + \frac{1}{r_{o_t}} \boldsymbol{v}_t. \end{aligned}$$

7: **end for**

An Accurate Online Implementation. Our strategy for solving $\boldsymbol{u}_t$ is based on an approximate routine which resolves Issue 4 as well as has a low complexity. Yet, to tackle Issue 4, another potential way is to avoid the variable $\boldsymbol{u}_t$[2]. Recall that we derive the optimal solution $\boldsymbol{U}^*$ (provided that $\boldsymbol{D}$ is given) to (2.4) is given by (2.12). Plugging it back to (2.4), we obtain

$$\begin{aligned} \|\boldsymbol{U}^*\|_F^2 &= \operatorname{Tr}\left(\boldsymbol{D}\boldsymbol{D}^\top\left(\boldsymbol{Q}_n - {\lambda_3}^{-1}\boldsymbol{Q}_n^2\right)\right), \\ \|\boldsymbol{D} - \boldsymbol{Y}\boldsymbol{U}^*\|_F^2 &= \left\|\boldsymbol{D} - {\lambda_3}^{-1}\boldsymbol{Q}_n\boldsymbol{D}\right\|_F^2, \end{aligned}$$

where

$$\boldsymbol{Q}_n = \left({\lambda_3}^{-1}\boldsymbol{I}_p + \boldsymbol{N}_n\right)^{-1}.$$

Here, $\boldsymbol{N}_n$ was given in (2.14). Note that the size of $\boldsymbol{Q}_n$ is $p \times p$. Hence, if we incrementally compute the accumulation matrix $\boldsymbol{N}_t = \sum_{i=1}^t \boldsymbol{y}_i\boldsymbol{y}_i^\top$, we can update the variable $\boldsymbol{D}$ in an online fashion. Namely, at t-th iteration, we re-define the surrogate function as follows:

$$g_t(\boldsymbol{D}) \overset{\text{def}}{=} \frac{1}{t}\left[\sum_{i=1}^t \tilde{\ell}(\boldsymbol{z}_i, \boldsymbol{D}, \boldsymbol{v}_i, \boldsymbol{e}_i) + \frac{\lambda_3}{2}\left\|\boldsymbol{D} - \frac{1}{\lambda_3}\boldsymbol{Q}_t\boldsymbol{D}\right\|_F^2 + \frac{1}{2}\operatorname{Tr}\left(\boldsymbol{D}\boldsymbol{D}^\top\left(\boldsymbol{Q}_t - \frac{1}{\lambda_3}\boldsymbol{Q}_t^2\right)\right)\right].$$

[2] We would like to thank the anonymous NIPS 2015 Reviewer for pointing out this potential solution to the online algorithm. Here we explain why this alternative can be computationally expensive.

Again, by noting the fact that $\tilde{\ell}(\boldsymbol{z}_i, \boldsymbol{D}, \boldsymbol{v}_i, \boldsymbol{e}_i)$ only involves recording $\boldsymbol{A}_t$ and $\boldsymbol{B}_t$, we show that the memory cost is independent of sample size.

While promising since the above procedure avoids the approximate computation, the main shortcoming is computing the inverse of a $p \times p$ matrix in each iteration, hence not efficient. Moreover, as we will show in Theorem 1, although the $\boldsymbol{u}_t$'s are approximate solutions, we are still guaranteed the convergence of $\boldsymbol{D}_t$.

4 Theoretical Analysis

We make three assumptions underlying our analysis.

Assumption 1. The observed data are generated i.i.d. from some distribution and there exist constants α_0 and α_1, such that the conditions $0 < \alpha_0 \leq \|\boldsymbol{z}\|_2 \leq \alpha_1$ and $\alpha_0 \leq \|\boldsymbol{y}\|_2 \leq \alpha_1$ hold almost surely.

Assumption 2. The smallest singular value of the matrix $\frac{1}{t}\boldsymbol{N}_t$ is lower bounded away from zero.

Assumption 3. The surrogate functions $g_t(\boldsymbol{D})$ are strongly convex for all $t \geq 0$.

Based on these assumptions, we establish the main theoretical result, justifying the validity of Algorithm 1.

Theorem 1. *Assume 1, 2 and 3. Let $\{\boldsymbol{D}_t\}_{t=1}^{\infty}$ be the sequence of optimal bases produced by Algorithm 1. Then, the sequence converges to a stationary point of the expected loss function $f(\boldsymbol{D})$ when t goes to infinity.*

Note that since the reformulation of the nuclear norm (2.1) is non-convex, we can only guarantee that the solution is a stationary point in general [Ber99]. We also remark that OLRSC asymptotically fulfills the first order optimality condition of (1.1). To see this, we follow the proof technique of Prop.3 in [MMG15] and let $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\top, W_1 = \boldsymbol{U}\boldsymbol{U}^\top, W_2 = \boldsymbol{V}\boldsymbol{V}^\top, \boldsymbol{M}_1 = \boldsymbol{M}_3 = 0.5\boldsymbol{I}, \boldsymbol{M}_2 = \boldsymbol{M}_4 = 0.5\lambda_1\boldsymbol{Y}^\top(\boldsymbol{Y}\boldsymbol{X} + \boldsymbol{E} - \boldsymbol{Z})$. Due to our uniform bound (Prop. 7), we justify the optimality condition. See the details in [MMG15].

More interestingly, as we mentioned in Section 3, the solution (3.5) is not accurate in the sense that it is not equal to that of (2.15) given $\boldsymbol{D}$. Yet, our theorem asserts that this will not deviate $\{\boldsymbol{D}_t\}_{t\geq 0}$ away from the stationary point. The intuition underlying such amazing phenomenon is that the expected loss function (2.20) is only determined by $\ell(\boldsymbol{z}, \boldsymbol{D})$ which does not involve $\boldsymbol{u}_t$. What is of matter for $\boldsymbol{u}_t$ and $\boldsymbol{M}_t$ is their uniform boundedness and concentration to establish the convergence. Thanks to the carefully chosen function $\tilde{\ell}(\boldsymbol{z}, \boldsymbol{D}, \boldsymbol{M}, \boldsymbol{u})$ and the surrogate function $g_t(\boldsymbol{D})$, we are able to prove the desired property by mathematical induction which is a crucial step in our proof.

In particular, we have the following lemma that facilitates our analysis:

Lemma 2. *Assume 1 and 2 and 3. Let $\{\boldsymbol{M}_t\}_{t\geq 0}$ be the sequence of the matrices produced by Algorithm 1. Then, there exists some universal constant C_0, such that for all $t \geq 0$, $\|M_t\|_F \leq \mathrm{C}_0$.*

Due to the above lemma, the solution $\boldsymbol{D}_t$ is essentially determined by $\frac{1}{t}\boldsymbol{A}_t$ and $\frac{1}{t}\boldsymbol{B}_t$ when t is a very large quantity since $\frac{1}{t}\boldsymbol{M}_t \to 0$. We also have a non-asymptotic rate for the numerical convergence of $\boldsymbol{D}_t$ as $\|\boldsymbol{D}_t - \boldsymbol{D}_{t-1}\|_2 = \mathcal{O}(1/t)$. See Appendix B for more details and a full proof.

5 Experiments

Before presenting the empirical results, we first introduce the universal settings used throughout the section.

Algorithms. For the subspace recovery task, we compare our algorithm with state-of-the-art solvers including ORPCA [FXY13], LRR [LLY+13] and PCP [CLMW11]. For the subspace clustering task, we choose

ORPCA, LRR and SSC [EV09] as the competitive baselines. Recently, [LL14] improved the vanilla LRR by utilizing some low-rank matrix for $\boldsymbol{Y}$. We denote this variant of LRR by LRR2 and accordingly, our algorithm equipped with such $\boldsymbol{Y}$ is denoted as OLRSC2.

Evaluation Metric. We evaluate the fitness of the recovered subspaces $\boldsymbol{D}$ (with each column being normalized) and the ground truth L by the Expressed Variance (EV) [XCM10]:

$$\mathrm{EV}(\boldsymbol{D}, \boldsymbol{L}) \overset{\mathrm{def}}{=} \frac{\mathrm{Tr}(\boldsymbol{D}\boldsymbol{D}^\top \boldsymbol{L}\boldsymbol{L}^\top)}{\mathrm{Tr}(\boldsymbol{L}\boldsymbol{L}^\top)}. \tag{5.1}$$

The value of EV scales between 0 and 1, and a higher value means better recovery.

The performance of subspace clustering is measured by clustering accuracy, which also ranges in the interval $[0, 1]$, and a higher value indicates a more accurate clustering.

Parameters. We set $\lambda_1 = 1$, $\lambda_2 = 1/\sqrt{p}$ and $\lambda_3 = \sqrt{t/p}$, where t is the iteration counter. These settings are actually used in ORPCA. We follow the default parameter setting for the baselines.

5.1 Subspace Recovery

Simulation Data. We use 4 disjoint subspaces $\{\mathcal{S}_k\}_{k=1}^4 \subset \mathbb{R}^p$, whose bases are denoted by $\{L_k\}_{k=1}^4 \in \mathbb{R}^{p \times d_k}$. The clean data matrix $\bar{\boldsymbol{Z}}_k \in \mathcal{S}_k$ is then produced by $\bar{\boldsymbol{Z}}_k = L_k R_k^\top$, where $R_k \in \mathbb{R}^{n_k \times d_k}$. The entries of L_k's and R_k's are sampled i.i.d. from the normal distribution. Finally, the observed data matrix $\boldsymbol{Z}$ is generated by $\boldsymbol{Z} = \bar{\boldsymbol{Z}} + \boldsymbol{E}$, where $\bar{\boldsymbol{Z}}$ is the column-wise concatenation of $\bar{\boldsymbol{Z}}_k$'s followed by a random permutation, $\boldsymbol{E}$ is the sparse corruption whose ρ fraction entries are non-zero and follow an i.i.d. uniform distribution over $[-2, 2]$. We independently conduct each experiment 10 times and report the averaged results.

Robustness. We illustrate by simulation results that OLRSC can effectively recover the underlying subspaces, confirming that $\boldsymbol{D}_t$ converges to the union of subspaces. For the two online algorithms OLRSC and ORPCA, We compute the EV after revealing all the samples. We examine the performance under different intrinsic dimension d_k's and corruption ρ. To be more detailed, the d_k's are varied from $0.01p$ to $0.1p$ with a step size $0.01p$, and the ρ is from 0 to 0.5, with a step size 0.05.

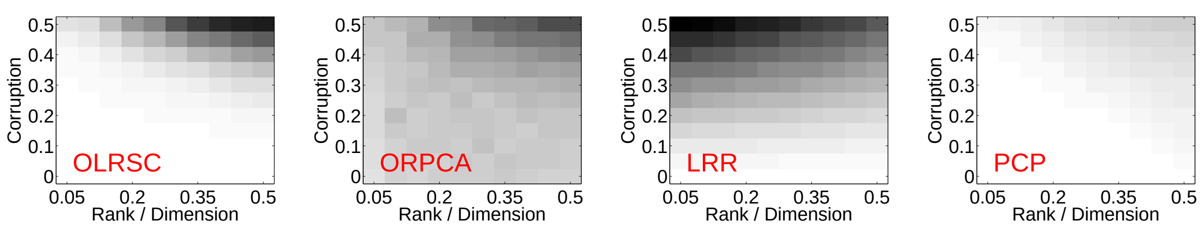

Figure 1: **Subspace recovery under different intrinsic dimensions and corruptions.** Brighter is better. We set $p = 100$, $n_k = 1000$ and $d = 4d_k$. LRR and PCP are batch methods. OLRSC consistently outperforms ORPCA and even improves the performance of LRR. Compared to PCP, OLRSC is competitive in most cases and degrades a little for highly corrupted data, possibly due to the number of samples is not sufficient for its convergence.

The results are presented in Figure 1. The most intriguing observation is that OLRSC as an online algorithm outperforms its batch counterpart LRR! Such improvement may come from the explicit modeling for the basis, which makes OLRSC more informative than LRR. Interestingly, [GQV14] also observed that in some situations, an online algorithm can outperform the batch counterpart. To fully understand the rationale

behind this phenomenon is an important direction for future research. Notably, OLRSC consistently beats ORPCA (an online version of PCP), which may be the consequence of the fact that OLRSC takes into account that the data are produced by a union of small subspaces. While PCP works well for almost all scenarios, OLRSC degrades a little when addressing difficult cases (high rank and corruption). This is not surprising since Theorem 1 is based on asymptotic analysis and hence, we expect that OLRSC will converge to the true subspace after acquiring more samples.

Convergence Rate. Now we test on a large dataset to show that our algorithm usually converges to the true subspace faster than ORPCA. We plot the EV curve against the number of samples in Figure 2. Firstly, when equipped with a proper matrix $\boldsymbol{Y}$, OLRSC2 and LRR2 can always produce an exact recovery of the subspace as PCP does. When using the dataset itself for $\boldsymbol{Y}$, OLRSC still converges to a favorable point after revealing all the samples. Compared to ORPCA, OLRSC is more robust and converges much faster for hard cases (see, e.g., $\rho = 0.5$). Again, we note that in such hard cases, OLRSC outperforms LRR, which agrees with the observation in Figure 1.

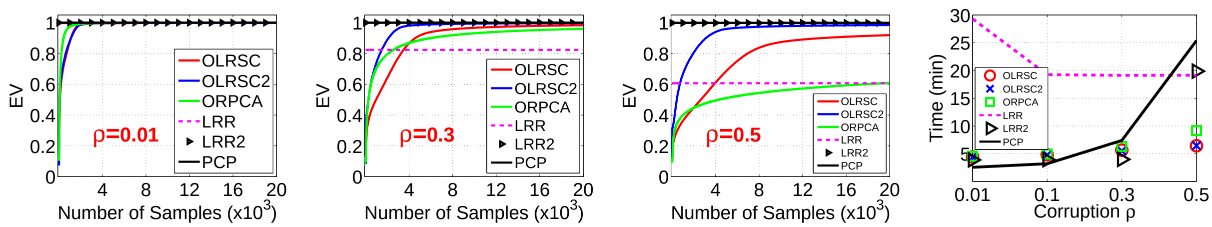

Figure 2: **Convergence rate and time complexity.** A higher EV means better subspace recovery. We set $p = 1000$, $n_k = 5000$, $d_k = 25$ and $d = 100$. OLRSC always converges to or outperforms the batch counterpart LRR. For hard cases, OLRSC converges much faster than ORPCA. Both PCP and LRR2 achieve the best EV value. When equipped with the same dictionary as LRR2, OLRSC2 also well handles the highly corrupted data ($\rho = 0.5$). Our methods are more efficient than the competitors but PCP when ρ is small, possibly because PCP utilizes a highly optimized C++ toolkit while ours are written in Matlab.

Computational Efficiency. We also illustrate the time complexity of the algorithms in the last panel of Figure 2. In short, our algorithms (OLRSC and OLRSC2) admit the lowest computational complexity for all cases. One may argue that PCP spends slightly less time than ours for a small ρ (0.01 and 0.1). However, we remark here that PCP utilizes a highly optimized C++ toolkit to boost computation while our algorithms are fully written in Matlab. We believe that ours will work more efficiently if properly optimized by, e.g., the blas routine. Another important message conveyed by the figure is that, OLRSC is always being orders of magnitude computationally more efficient than the batch method LRR, as well as producing comparable or even better solution.

5.2 Subspace Clustering

Datasets. We examine the performance for subspace clustering on 5 realistic databases shown in Table 1, which can be downloaded from the LibSVM website. For MNIST, We randomly select 20000 samples to form MNIST-20K since we find it time consuming to run the batch methods on the entire database.

Standard Clustering Pipeline. In order to focus on the solution quality of different algorithms, we follow the standard pipeline which feeds $\boldsymbol{X}$ to a spectral clustering algorithm [NJW01]. To this end, we collect all the $\boldsymbol{u}$'s and $\boldsymbol{v}$'s produced by OLRSC to form the representation matrix $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\top$. For ORPCA, we use $\boldsymbol{R}_0\boldsymbol{R}_0^\top$ as the similarity matrix [LLY+13], where $\boldsymbol{R}_0$ is the row space of $\boldsymbol{Z}_0 = \boldsymbol{L}_0\boldsymbol{\Sigma}_0\boldsymbol{R}_0^\top$ and $\boldsymbol{Z}_0$ is

Table 1: **Datasets for subspace clustering.**

	#classes	#samples	#features
Mushrooms	2	8124	112
DNA	3	3186	180
Protein	3	24,387	357
USPS	10	9298	256
MNIST-20K	10	20,000	784

the clean matrix recovered by ORPCA. We run our algorithm and ORPCA with 2 epochs so as to apply backward correction on the coefficients ($\boldsymbol{U}$ and $\boldsymbol{V}$ in ours and $\boldsymbol{R}_0$ in ORPCA).

Fully Online Pipeline. As we discussed in Section 3, the (optional) spectral clustering procedure needs the similarity matrix $\boldsymbol{X}$, making the memory proportional to n^2. To tackle this issue, we proposed a fully online scheme where the key idea is performing k-means on $\boldsymbol{V}$. Here, we examine the efficacy of this variant, which is called OLRSC-F.

Table 2: **Clustering accuracy (%) and computational time (seconds).** For each dataset, the first row indicates the accuracy and the second row the running time. For all the large-scale datasets, OLRSC (or OLRSC-F) has the highest clustering accuracy. Regarding the running time, our method spends comparable time as ORPCA (the fastest solver) does while dramatically improves the accuracy. Although SSC is slightly better than SSC on Protein, it consumes one hour while OLRSC takes 25 seconds.

	OLRSC	OLRSC-F	ORPCA	LRR	SSC
Mush-rooms	85.09 8.78	**89.36** 8.78	65.26 8.30	58.44 46.82	54.16 32 min
DNA	67.11 2.58	**83.08** 2.58	53.11 2.09	44.01 23.67	52.23 3 min
Protein	43.30 24.66	43.94 24.66	40.22 22.90	40.31 921.58	**44.27** 65 min
USPS	65.95 33.93	**70.29** 33.93	55.70 27.01	52.98 257.25	47.58 50 min
MNIST-20K	**57.74** 129	55.50 129	54.10 121	55.23 32 min	43.91 7 hours

The results are recorded in Table 2, where the time cost of spectral clustering or k-means is not included so we can focus on comparing the efficiency of the algorithms themselves. Also note that we use the dataset itself as the dictionary $\boldsymbol{Y}$ because we find that an alternative choice of $\boldsymbol{Y}$ does not help much on this task. For OLRSC and ORPCA, they require an estimation on the true rank. Here, we use $5k$ as such estimation where k is the number of classes of a dataset. Our algorithm significantly outperforms the two state-of-the-art methods LRR and SSC both for accuracy and efficiency. One may argue that SSC is slightly better than OLRSC on Protein. Yet, it spends 1 hour while OLRSC only costs 25 seconds. Hence, SSC is not practical. Compared to ORPCA, OLRSC always identifies more correct samples as well as consumes comparable running time. For example, on the USPS dataset, OLRSC achieves the accuracy of 65.95% while that of ORPCA is 55.7%. Regarding the running time, OLRSC uses only 7 seconds more than ORPCA – same order of computational complexity, which agrees with the qualitative analysis in Section 3.

Table 3: **Time cost (seconds) of spectral clustering and k-means.**

	Mushrooms	DNA	Protein	USPS	MNIST-20K
Spectral	295	18	7567	482	4402
k-means	2	6	5	19	91

More interestingly, it shows that the k-means alternative (OLRSC-F) usually outperforms the spectral clustering pipeline. This suggests that perhaps for *robust* subspace clustering formulations, the simple k-means paradigm suffices to guarantee an appealing result. On the other hand, we report the running time of spectral clustering and k-means in Table 3. As expected, since spectral clustering computes SVD for an n-by-n similarity matrix, it is quite slow. In fact, it sometimes dominates the running time of the whole pipeline. In contrast, k-means is extremely fast and scalable, as it can be implemented in online fashion.

5.3 Influence of d

A key ingredient of our formulation is a factorization on the nuclear norm regularized matrix, which requires an estimation on the rank of the $\boldsymbol{X}$ (see (2.1)). Here we examine the influence of the selection of d (which plays as an upper bound of the true rank). We report both EV and clustering accuracy for different d under a range of corruptions. The simulation data are generated as in Section 5.1 and we set $p = 200$, $n_k = 1000$ and $d_k = 10$. Since the four subspaces are disjoint, the true rank is 40.

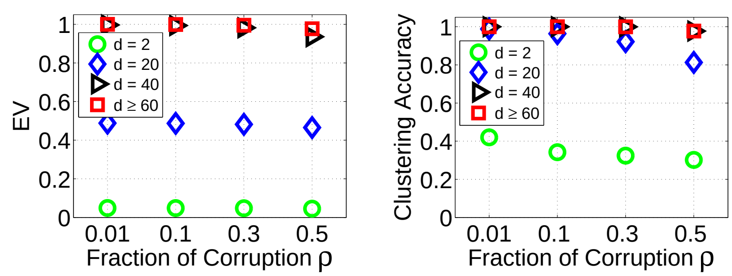

Figure 3: **Examine the influence of d.** We experiment on $d = \{2, 20, 40, 60, 80, 100, 120, 140, 160, 180\}$. The true rank is 40.

From Figure 3, we observe that our algorithm cannot recover the true subspace if d is smaller than the true rank. On the other hand, when d is sufficiently large (at least larger than the true rank), our algorithm can perfectly estimate the subspace. This agrees with the results in [BM05] which says as long as d is large enough, any local minima is global optima. We also illustrate the influence of d on subspace clustering. Generally speaking, OLRSC can consistently identify the cluster of the data points if d is sufficiently large. Interestingly, different from the subspace recovery task, here the requirement for d seems to be slightly relaxed. In particular, we notice that if we pick d as 20 (smaller than the true rank), OLRSC still performs well. Such relaxed requirement of d may benefit from the fact that the spectral clustering step can correct some wrong points as suggested by [SEC14].

6 Conclusion

In this paper, we have proposed an online algorithm called OLRSC for subspace clustering, which dramatically reduces the memory cost of LRR from $\mathcal{O}(n^2)$ to $\mathcal{O}(pd)$ – orders of magnitudes more memory efficient.

One of the key techniques in this work is an explicit basis modeling, which essentially renders the model more informative than LRR. Another important component is a non-convex reformulation of the nuclear norm. Combining these techniques allows OLRSC to simultaneously recover the union of the subspaces, identify the possible corruptions and perform subspace clustering. We have also established the theoretical guarantee that solutions produced by our algorithm converge to a stationary point of the expected loss function. Moreover, we have analyzed the time complexity and empirically demonstrated that our algorithm is computationally very efficient compared to competing baselines. Our extensive experimental study on synthetic and realistic datasets also illustrates the robustness of OLRSC. In a nutshell, OLRSC is an appealing algorithm in all three worlds: memory cost, computation and robustness.

A Proof Preliminaries

Lemma 3 (Corollary of Thm. 4.1 [BS98]). *Let $f : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$. Suppose that for all $\boldsymbol{x} \in \mathbb{R}^p$ the function $f(\boldsymbol{x}, \cdot)$ is differentiable, and that f and $\nabla_{\boldsymbol{u}} f(\boldsymbol{x}, \boldsymbol{u})$ are continuous on $\mathbb{R}^p \times \mathbb{R}^q$. Let $\boldsymbol{v}(\boldsymbol{u})$ be the optimal value function $\boldsymbol{v}(\boldsymbol{u}) = \min_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x}, \boldsymbol{u})$, where $\mathcal{C}$ is a compact subset of $\mathbb{R}^p$. Then $\boldsymbol{v}(\boldsymbol{u})$ is directionally differentiable. Furthermore, if for $\boldsymbol{u}_0 \in \mathbb{R}^q$, $f(\cdot, \boldsymbol{u}_0)$ has unique minimizer $\boldsymbol{x}_0$ then $\boldsymbol{v}(\boldsymbol{u})$ is differentiable in $\boldsymbol{u}_0$ and $\nabla_{\boldsymbol{u}} \boldsymbol{v}(\boldsymbol{u}_0) = \nabla_{\boldsymbol{u}} f(\boldsymbol{x}_0, \boldsymbol{u}_0)$.*

Lemma 4 (Corollary of Donsker theorem [vdV00]). *Let $F = \{f_\theta : \mathcal{X} \to \mathbb{R}, \theta \in \Theta\}$ be a set of measurable functions indexed by a bounded subset Θ of $\mathbb{R}^d$. Suppose that there exists a constant K such that*

$$|f_{\theta_1}(x) - f_{\theta_2}(x)| \leq K \left\|\theta_1 - \theta_2\right\|_2,$$

for every θ_1 and θ_2 in Θ and x in $\mathcal{X}$. Then, F is P-Donsker. For any f in F, let us define $\mathbb{P}_n f$, $\mathbb{P} f$ and $\mathbb{G}_n f$ as

$$\begin{aligned}
\mathbb{P}_n f &= \frac{1}{n} \sum_{i=1}^{n} f(X_i), \\
\mathbb{P} f &= \mathbb{E}[f(X)], \\
\mathbb{G}_n f &= \sqrt{n}(\mathbb{P}_n f - \mathbb{P} f).
\end{aligned}$$

Let us also suppose that for all f, $\mathbb{P} f^2 < \delta^2$ and $\|f\|_\infty < \boldsymbol{M}$ and that the random elements $X_1, X_2, \cdots$ are Borel-measurable. Then, we have

$$\mathbb{E} \left\|\mathbb{G}\right\|_F = O(1),$$

where $\left\|\mathbb{G}\right\|_F = \sup_{f \in F} |\mathbb{G}_n f|$.

Lemma 5 (Sufficient condition of convergence for a stochastic process [Bot98]). *Let $(\Omega, \mathcal{F}, P)$ be a measurable probability space, u_t, for $t \geq 0$, be the realization of a stochastic process and $\mathcal{F}_t$ be the filtration by the past information at time t. Let*

$$\delta_t = \begin{cases} 1 & if \ \mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t] > 0, \\ 0 & otherwise. \end{cases}$$

If for all t, $u_t \geq 0$ and $\sum_{t=1}^{\infty} \mathbb{E}[\delta_t (u_{t+1} - u_t)] < \infty$, then u_t is a quasi-martingale and converges almost surely. Moreover,

$$\sum_{t=1}^{\infty} |\mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t]| < +\infty \ a.s.$$

Lemma 6 (Lemma 8 from [MBPS10]). *Let a_t, b_t be two real sequences such that for all t, $a_t \geq 0$, $b_t \geq 0$, $\sum_{t=1}^{\infty} a_t = \infty$, $\sum_{t=1}^{\infty} a_t b_t < \infty$, $\exists K > 0$, such that $|b_{t+1} - b_t| < K a_t$. Then, $\lim_{t \to +\infty} b_t = 0$.*

B Proof Details

B.1 Proof of Boundedness

Proposition 7. *Let $\{\boldsymbol{u}_t\}$, $\{\boldsymbol{v}_t\}$, $\{\boldsymbol{e}_t\}$ and $\{\boldsymbol{D}_t\}$ be the optimal solutions produced by Algorithm 1. Then,*

1. *$\boldsymbol{v}_t$, $\boldsymbol{e}_t$, $\frac{1}{t}\boldsymbol{A}_t$ and $\frac{1}{t}\boldsymbol{B}_t$ are uniformly bounded.*

2. *$\boldsymbol{M}_t$ is uniformly bounded.*

3. *$\boldsymbol{D}_t$ is supported by some compact set $\mathcal{D}$.*

4. *$\boldsymbol{u}_t$ is uniformly bounded.*

Proof. Let us consider the optimization problem of solving $\boldsymbol{v}$ and $\boldsymbol{e}$. As the trivial solution $\{\boldsymbol{v}_t', \boldsymbol{e}_t'\} = \{\boldsymbol{0}, \boldsymbol{z}_t\}$ are feasible, we have

$$\tilde{\ell}_1(\boldsymbol{z}_t, \boldsymbol{D}_{t-1}, \boldsymbol{v}_t', \boldsymbol{e}_t') = \lambda_2 \left\| \boldsymbol{z}_t \right\|_1.$$

Therefore, the optimal solution should satisfy:

$$\frac{\lambda_1}{2} \left\| \boldsymbol{z}_t - \boldsymbol{D}_{t-1}\boldsymbol{v}_t - \boldsymbol{e}_t \right\|_2^2 + \frac{1}{2} \left\| \boldsymbol{v}_t \right\|_2^2 + \lambda_2 \left\| \boldsymbol{e}_t \right\|_1 \leq \lambda_2 \left\| \boldsymbol{z}_t \right\|_1,$$

which implies

$$\frac{1}{2} \left\| \boldsymbol{v}_t \right\|_2^2 \leq \lambda_2 \left\| \boldsymbol{z}_t \right\|_1, \quad \lambda_2 \left\| \boldsymbol{e}_t \right\|_1 \leq \lambda_2 \left\| \boldsymbol{z}_t \right\|_1.$$

Since $\boldsymbol{z}_t$ is uniformly bounded (Assumption 1), $\boldsymbol{v}_t$ and $\boldsymbol{e}_t$ are uniformly bounded.

To examine the uniform bound for $\frac{1}{t}\boldsymbol{A}_t$ and $\frac{1}{t}\boldsymbol{B}_t$, note that

$$\frac{1}{t}\boldsymbol{A}_t = \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{v}_i \boldsymbol{v}_i^\top,$$
$$\frac{1}{t}\boldsymbol{B}_t = \frac{1}{t}\sum_{i=1}^{t} (\boldsymbol{z}_i - \boldsymbol{e}_i)\, \boldsymbol{v}_i^\top.$$

Since for each i, $\boldsymbol{v}_i$, $\boldsymbol{e}_i$ and $\boldsymbol{z}_i$ are uniformly bounded, $\frac{1}{t}\boldsymbol{A}_t$ and $\frac{1}{t}\boldsymbol{B}_t$ are uniformly bounded.

Now we derive the bound for $\boldsymbol{M}_t$. All the information we have is:

1. $\boldsymbol{M}_t = \sum_{i=1}^t \boldsymbol{y}_i \boldsymbol{u}_i^\top$ (definition of $\boldsymbol{M}_t$).
2. $\boldsymbol{u}_t = (\|\boldsymbol{y}_t\|_2^2 + \frac{1}{\lambda_3})^{-1}(\boldsymbol{D}_{t-1} - \boldsymbol{M}_{t-1})^\top \boldsymbol{y}_t$ (closed form solution).
3. $\boldsymbol{D}_t(\lambda_1 \boldsymbol{A}_t + \lambda_3 \boldsymbol{I}) = \lambda_1 \boldsymbol{B}_t + \lambda_3 \boldsymbol{M}_t$ (first order optimality condition for $\boldsymbol{D}_t$).
4. $\frac{1}{t}\boldsymbol{A}_t$, $\frac{1}{t}\boldsymbol{B}_t$, $\frac{1}{t}\boldsymbol{N}_t$ are uniformly upper bounded (Claim 1).
5. The smallest singular values of $\frac{1}{t}\boldsymbol{N}_t$ and $\frac{1}{t}\boldsymbol{A}_t$ are uniformly lower bounded away from zero (Assumption 2 and 3).

For simplicity, we write $\boldsymbol{D}_t$ as:

$$\boldsymbol{D}_t = (\lambda_1 \boldsymbol{B}_t + \lambda_3 \boldsymbol{M}_t)\boldsymbol{Q}_t^{-1}, \tag{B.1}$$

where

$$\boldsymbol{Q}_t = \lambda_1 \boldsymbol{A}_t + \lambda_3 \boldsymbol{I}.$$

Note that as we assume $\frac{1}{t}\boldsymbol{A}_t$ is positive definite, $\boldsymbol{Q}_t$ is always invertible.

From the definition of $\boldsymbol{M}_t$ and (3.5), we know that

$$\begin{aligned}\boldsymbol{M}_{t+1} - \boldsymbol{M}_t &= \boldsymbol{y}_{t+1}\boldsymbol{u}_{t+1}^\top \\ &= \left(\|\boldsymbol{y}_{t+1}\|_2^2 + \frac{1}{\lambda_3}\right)^{-1} \boldsymbol{y}_{t+1}\boldsymbol{y}_{t+1}^\top (\boldsymbol{D}_t - \boldsymbol{M}_t) \\ &= \boldsymbol{P}_t\boldsymbol{D}_t - \boldsymbol{P}_t\boldsymbol{M}_t \\ &= \boldsymbol{P}_t(\lambda_1\boldsymbol{B}_t + \lambda_3\boldsymbol{M}_t)\boldsymbol{Q}_t^{-1} - \boldsymbol{P}_t\boldsymbol{M}_t, \end{aligned} \tag{B.2}$$

where

$$\boldsymbol{P}_t = \left(\|\boldsymbol{y}_{t+1}\|_2^2 + \frac{1}{\lambda_3}\right)^{-1} \boldsymbol{y}_{t+1}\boldsymbol{y}_{t+1}^\top.$$

By multiplying $\boldsymbol{Q}_t$ on both sides of (B.2), we have

$$\boldsymbol{M}_{t+1} = \left(\boldsymbol{M}_t - \lambda_1\boldsymbol{P}_t\boldsymbol{M}_t\boldsymbol{A}_t\boldsymbol{Q}_t^{-1}\right) + \lambda_1\boldsymbol{P}_t\boldsymbol{B}_t\boldsymbol{Q}_t^{-1}. \tag{B.3}$$

By applying the Taylor expansion on $\boldsymbol{Q}_t^{-1}$, we have

$$\boldsymbol{Q}_t^{-1} = (\lambda_1\boldsymbol{A}_t + \lambda_3\boldsymbol{I}_d)^{-1} = \frac{1}{\lambda_3}\sum_{i=0}^{+\infty}\left(-\frac{\lambda_1}{\lambda_3}\boldsymbol{A}_t\right)^i.$$

Thus,

$$\begin{aligned}\boldsymbol{A}_t\boldsymbol{Q}_t^{-1} &= \frac{1}{\lambda_3}\sum_{i=0}^{+\infty}\left(-\frac{\lambda_1}{\lambda_3}\right)^i (\boldsymbol{A}_t)^{i+1} \\ &= -\frac{1}{\lambda_1}\sum_{i=0}^{+\infty}\left(-\frac{\lambda_1}{\lambda_3}\boldsymbol{A}_t\right)^{i+1} \\ &= -\frac{1}{\lambda_1}\left[\sum_{i=-1}^{+\infty}\left(-\frac{\lambda_1}{\lambda_3}\boldsymbol{A}_t\right)^{i+1} - \boldsymbol{I}_d\right] \\ &= -\frac{1}{\lambda_1}\left(\boldsymbol{I}_d + \frac{\lambda_1}{\lambda_3}\boldsymbol{A}_t\right)^{-1} + \frac{1}{\lambda_1}\boldsymbol{I}_d. \end{aligned}$$

So $\boldsymbol{M}_{t+1}$ is given by

$$\boldsymbol{M}_{t+1} = (\boldsymbol{I}_d - \boldsymbol{P}_t)\boldsymbol{M}_t + \underbrace{\boldsymbol{P}_t\boldsymbol{M}_t\left(\boldsymbol{I}_d + \frac{\lambda_1}{\lambda_3}\boldsymbol{A}_t\right)^{-1} + \lambda_1\boldsymbol{P}_t\boldsymbol{B}_t\boldsymbol{Q}_t^{-1}}_{\boldsymbol{W}_t}. \tag{B.4}$$

We first show that $\boldsymbol{P}_t\boldsymbol{B}_t\boldsymbol{Q}_t^{-1}$ is uniformly bounded.

$$\|\boldsymbol{P}_t\boldsymbol{B}_t\boldsymbol{Q}_t^{-1}\| = \left\|\boldsymbol{P}_t\left(\frac{1}{t}\boldsymbol{B}_t\right)\left(\frac{1}{t}\boldsymbol{Q}_t\right)^{-1}\right\| \le \|\boldsymbol{P}_t\| \cdot \left\|\frac{1}{t}\boldsymbol{B}_t\right\| \cdot \left\|\left(\frac{1}{t}\boldsymbol{Q}_t\right)^{-1}\right\|.$$

Since we assume that $\{\boldsymbol{z}_t\}$ are uniformly upper bounded (Assumption 1), there exists a constant α_1, such that for all $t > 0$,

$$\|\boldsymbol{z}_t\|_2 \le \alpha_1.$$

So we have

$$\|\boldsymbol{P}_{t+1}\| \leq \frac{\lambda_3\alpha_1^2}{\lambda_3\alpha_1^2+1}.$$

Next, as we have shown that $\frac{1}{t}\boldsymbol{B}_t$ can be uniformly bounded, there exists a constant c_1, such that for all $t>0$,

$$\left\|\frac{1}{t}\boldsymbol{B}_t\right\| \leq c_1.$$

And,

$$\begin{aligned}
\left\|\left(\frac{1}{t}\boldsymbol{Q}_t\right)^{-1}\right\| &= \frac{1}{\sigma_{\min}\left(\frac{1}{t}\boldsymbol{Q}_t\right)} \\
&= \frac{1}{\sigma_{\min}\left(\frac{\lambda_1}{t}\boldsymbol{A}_t+\frac{\lambda_3}{t}\boldsymbol{I}_d\right)} \\
&= \frac{1}{\frac{\lambda_3}{t}+\lambda_1\sigma_{\min}\left(\frac{1}{t}\boldsymbol{A}_t\right)} \\
&\leq \frac{1}{\lambda_3+\lambda_1\beta_0}.
\end{aligned}$$

Thus, $\lambda_1\boldsymbol{P}_t\boldsymbol{B}_t\boldsymbol{Q}_t^{-1}$ is uniformly bounded by a constant, say c_2. That is,

$$\left\|\lambda_1\boldsymbol{P}_t\boldsymbol{B}_t\boldsymbol{Q}_t^{-1}\right\| \leq c_2. \tag{B.5}$$

It follows that $\boldsymbol{W}_t$ can be bounded:

$$\begin{aligned}
\|\boldsymbol{W}_t\| &\leq \|\boldsymbol{P}_t\|\cdot\|\boldsymbol{M}_t\|\cdot\left\|\left(\boldsymbol{I}_d+\frac{\lambda_1}{\lambda_3}\boldsymbol{A}_t\right)^{-1}\right\|+c_2 \\
&\overset{\zeta_1}{\leq} \frac{\alpha_1^2}{\alpha_1^2+\frac{1}{\lambda_3}}\cdot\frac{\lambda_3}{\lambda_3+\lambda_1\beta_0 t}\|\boldsymbol{M}_t\|+c_2 \\
&\leq \frac{c_3}{t}\|\boldsymbol{M}_t\|+c_2,
\end{aligned} \tag{B.6}$$

where ζ_1 is derived by utilizing the assumption that $\boldsymbol{z}$ is upper bounded by α_1 and the smallest singular value of $\frac{1}{t}\boldsymbol{A}_t$ is lower bounded by β_0. The last inequality always holds for some uniform constant c_3.

From Assumption 2, we know that the singular values of $\frac{1}{t}\sum_{i=1}^{t}\boldsymbol{z}_i\boldsymbol{z}_i^\top$ should uniformly span the diagonal. Thus, there exists a constant τ, such that for all $i>0$, $\frac{1}{\tau}\sum_{i}^{i+\tau}\boldsymbol{z}_i\boldsymbol{z}_i^\top$ is uniformly bounded away from zero with high probability.

Let $m_1=\|\boldsymbol{M}_1\|$. Now we pick a constant t^*, such that

$$\frac{c_3\tau}{t^*}\left(\frac{1}{\alpha_0}+1\right) \leq 0.5. \tag{B.7}$$

We also have a constant w^*, such that for all $t\leq t^*$,

$$\|\boldsymbol{W}_t\| \leq w^*, \quad \frac{c_3}{t}m_1+0.5w^*+c_2 \leq w^*. \tag{B.8}$$

Based on this, we first derive a bound for all $\|\boldsymbol{M}_t\|$, with $t \leq t^*$. We know that there exists an integer k^* (which is a uniform constant), such that $k^*(\tau+1) \leq t^* < (k^*+1)(\tau+1)$. Our strategy is to bound $\|\boldsymbol{M}_t\|$ in each interval $[(k-1)(\tau+1), k(\tau+1)]$. We start our reasoning from the first interval $[1, \tau+1]$.

It is easy to induce from (B.4) that for all $t > 0$,

$$\boldsymbol{M}_{t+1} = \prod_{i=1}^{t}(\boldsymbol{I}_p - \boldsymbol{P}_i)\boldsymbol{M}_1 + \sum_{j=1}^{t-1}\prod_{i=j+1}^{t}(\boldsymbol{I}_p - \boldsymbol{P}_i)\boldsymbol{W}_j + \boldsymbol{W}_t.$$

Thus,

$$\begin{aligned}
\|\boldsymbol{M}_{\tau+1}\| &= \left\|\prod_{i=1}^{\tau}(\boldsymbol{I}_p - \boldsymbol{P}_i)\boldsymbol{M}_1 + \sum_{j=1}^{\tau-1}\prod_{i=j+1}^{\tau}(\boldsymbol{I}_p - \boldsymbol{P}_i)\boldsymbol{W}_j + \boldsymbol{W}_\tau\right\| \\
&\leq \left\|\prod_{i=1}^{\tau}(\boldsymbol{I}_p - \boldsymbol{P}_i)\boldsymbol{M}_1\right\| + \left\|\sum_{j=1}^{\tau-1}\prod_{i=j+1}^{\tau}(\boldsymbol{I}_p - \boldsymbol{P}_i)\boldsymbol{W}_j + \boldsymbol{W}_\tau\right\| \\
&\overset{\zeta_1}{\leq} \left\|\prod_{i=1}^{\tau}(\boldsymbol{I}_p - \boldsymbol{P}_i)\right\| \cdot \|\boldsymbol{M}_1\| + \tau w^* \\
&\overset{\zeta_2}{\leq} (1-\alpha_0)m_1 + \tau w^*.
\end{aligned}$$

Here, ζ_1 holds because $\left\|\prod_{i=j+1}^{\tau}(\boldsymbol{I}_p - \boldsymbol{P}_i)\right\| \leq 1$ for all $j \in [\tau - 1]$. ζ_2 holds because the singular values of $\boldsymbol{P}_i$'s have span over the diagonal so the largest singular value of $\prod_{i=1}^{\tau}(\boldsymbol{I}_p - \boldsymbol{P}_i)$ is $1-\alpha_0$, where α_0 is the lower bound for all $\boldsymbol{z}_i$'s (see Assumption 1).

For $\boldsymbol{M}_{2(\tau+1)}$, we can similarly obtain

$$\left\|\boldsymbol{M}_{2(\tau+1)}\right\| \leq (1-\alpha_0)^2 m_1 + (1-\alpha_0)\tau w^* + \tau w^*.$$

More generally, for any integer $k \leq k^*$,

$$\left\|\boldsymbol{M}_{k(\tau+1)}\right\| \leq (1-\alpha_0)^k m_1 + \sum_{j=0}^{k-1}(1-\alpha_0)^j \tau w^* \leq m_1 + \frac{\tau w^*}{\alpha_0}.$$

Hence, we obtain a uniform bound for $\left\|\boldsymbol{M}_{k(\tau+1)}\right\|$, with $k \in [k^*]$. For any $i \in ((k-1)(\tau+1), k(\tau+1))$, they can simply bounded by

$$\|\boldsymbol{M}_i\| \leq m_1 + \frac{\tau w^*}{\alpha_0} + (i - (k-1)(\tau+1))w^* \leq m_1 + \frac{\tau w^*}{\alpha_0} + \tau w^*.$$

Therefore, for all the current $\boldsymbol{M}_t$'s, we can bound them as follows:

$$\|\boldsymbol{M}_t\| \leq m_1 + \frac{\tau w^*}{\alpha_0} + \tau w^*, \ \forall\, t = 1, 2, \cdots, t^*. \tag{B.9}$$

From (B.8) and (B.9), we know that for all $t \leq t^*$,

$$\|\boldsymbol{W}_t\| \leq w^*, \quad \|\boldsymbol{M}_t\| \leq m_1 + \frac{\tau w^*}{\alpha_0} + \tau w^*.$$

Next, we show that the bounds still hold for $\|\boldsymbol{W}_{t^*+1}\|$ and $\|\boldsymbol{M}_{t^*+1}\|$, which completes our induction.

For $\|\boldsymbol{M}_{t^*+1}\|$, it can simply be bounded in the same way as aforementioned because all the $\boldsymbol{W}_t$'s are bounded by w^* for $t < t^* + 1$. That is,

$$\|\boldsymbol{M}_{t^*+1}\| \leq \left\|\boldsymbol{M}_{k^*(\tau+1)}\right\| + (t^* + 1 - k^*(\tau+1))w^* \leq m_1 + \frac{\tau w^*}{\alpha_0} + \tau w^*. \tag{B.10}$$

For $\|\boldsymbol{W}_{t^*+1}\|$, from (B.6), we know

$$\begin{aligned}
\|\boldsymbol{W}_{t^*+1}\| &\leq \frac{c_3}{t^*+1}\|\boldsymbol{M}_{t^*+1}\| + c_2 \\
&\leq \frac{c_3}{t^*+1}(m_1 + \frac{\tau w^*}{\alpha_0} + \tau w^*) + c_2 \\
&= \frac{c_3 m_1}{t^*+1} + \frac{c_3 \tau}{t^*+1}(\frac{1}{\alpha_0} + 1)w^* + c_2 \\
&\overset{\zeta_1}{\leq} \frac{c_3 m_1}{t^*+1} + 0.5w^* + c_2 \\
&\overset{\zeta_2}{\leq} w^*.
\end{aligned} \tag{B.11}$$

Here, ζ_1 is derived by utilizing (B.7) and ζ_2 is derived by (B.8).

From (B.10) and (B.11), we know that the bound for $\|\boldsymbol{M}_t\|$ and $\|\boldsymbol{W}_t\|$'s, with $t \leq t^*$, still holds for $t = t^* + 1$. Thus we complete the induction and conclude that for all $t > 0$, we have

$$\|\boldsymbol{M}_t\| \leq m_1 + \frac{\tau w^*}{\alpha_0} + \tau w^*, \quad \|\boldsymbol{W}_t\| \leq w^*.$$

Thus, $\boldsymbol{M}_t$ is uniformly bounded.

From (B.1), we know that

$$\begin{aligned}
\boldsymbol{D}_t &= \lambda_1 \boldsymbol{B}_t \left(\lambda_1 \boldsymbol{A}_t + \lambda_3 \boldsymbol{I}_d\right)^{-1} + \lambda_3 \boldsymbol{M}_t \left(\lambda_1 \boldsymbol{A}_t + \lambda_3 \boldsymbol{I}_d\right)^{-1} \\
&= \lambda_1 \left(\frac{1}{t}\boldsymbol{B}_t\right)\left(\frac{\lambda_1}{t}\boldsymbol{A}_t + \frac{\lambda_3}{t}\boldsymbol{I}_d\right)^{-1} + \frac{\lambda_3}{t}\boldsymbol{M}_t\left(\frac{\lambda_1}{t}\boldsymbol{A}_t + \frac{\lambda_3}{t}\boldsymbol{I}_d\right)^{-1}.
\end{aligned}$$

Since $\frac{1}{t}\boldsymbol{A}_t$, $\frac{1}{t}\boldsymbol{B}_t$ and $\boldsymbol{M}_t$ are all uniformly bounded, $\boldsymbol{D}_t$ is also uniformly bounded.

By examining the closed form of $\boldsymbol{u}_t$, and note that we have proved the uniform boundedness of $\boldsymbol{D}_t$ and $\boldsymbol{M}_t$, we conclude that $\{\boldsymbol{u}_t\}$ are uniformly bounded. $\square$

Corollary 8. *Let $\boldsymbol{v}_t$, $\boldsymbol{e}_t$, $\boldsymbol{u}_t$ and $\boldsymbol{D}_t$ be the optimal solutions produced by Algorithm 1.*

1. *$\tilde{\ell}(\boldsymbol{z}_t, \boldsymbol{D}_t, \boldsymbol{v}_t, \boldsymbol{e}_t)$ and $\ell(\boldsymbol{z}_t, \boldsymbol{D}_t)$ are uniformly bounded.*
2. *$\frac{1}{t}\tilde{h}(\boldsymbol{Z}, \boldsymbol{D}, \boldsymbol{U})$ is uniformly bounded.*
3. *The surrogate function $g_t(\boldsymbol{D}_t)$ defined in (3.6) is uniformly bounded and Lipschitz.*

Proof. To show Claim 1, we just need to examine the definition of $\tilde{\ell}(\boldsymbol{z}_t, \boldsymbol{D}_t, \boldsymbol{v}_t, \boldsymbol{e}_t)$ and notice that $\boldsymbol{z}_t$, $\boldsymbol{D}_t$, $\boldsymbol{v}_t$ and $\boldsymbol{e}_t$ are all uniformly bounded. This implies that $\tilde{\ell}(\boldsymbol{z}_t, \boldsymbol{D}_t, \boldsymbol{v}_t, \boldsymbol{e}_t)$ is uniformly bounded and so is $\ell(\boldsymbol{z}_t, \boldsymbol{D}_t)$. Likewise, we show that $\frac{1}{t}\tilde{h}(\boldsymbol{Z}, \boldsymbol{D}, \boldsymbol{U})$ is uniformly bounded.

The uniform boundedness of $g_t(\boldsymbol{D}_t)$ follows immediately as $\tilde{\ell}(\boldsymbol{z}_t, \boldsymbol{D}_t, \boldsymbol{v}_t, \boldsymbol{e}_t)$ and $\frac{1}{t}\tilde{h}(\boldsymbol{Z}, \boldsymbol{D}, \boldsymbol{U})$ are both uniformly bounded. To show that $g_t(\boldsymbol{D})$ is Lipschitz, we show that the gradient of $g_t(\boldsymbol{D})$ is uniformly

bounded for all $\boldsymbol{D} \in \mathcal{D}$.

$$\|\nabla g_t(\boldsymbol{D})\|_F = \left\| \lambda_1 \boldsymbol{D} \left(\frac{\boldsymbol{A}_t}{t} + \frac{\lambda_3}{t} \boldsymbol{I}_d \right) - \lambda_1 \frac{\boldsymbol{B}_t}{t} - \frac{\lambda_3}{t} \boldsymbol{M}_t \right\|_F$$
$$\leq \lambda_1 \|\boldsymbol{D}\|_F \left(\left\| \frac{\boldsymbol{A}_t}{t} \right\|_F + \left\| \frac{\lambda_3}{t} \boldsymbol{I}_d \right\|_F \right) + \lambda_1 \left\| \frac{\boldsymbol{B}_t}{t} \right\|_F + \left\| \frac{\lambda_3}{t} \boldsymbol{M}_t \right\|_F .$$

Notice that each term on the right side of the inequality is uniformly bounded. Thus the gradient of $g_t(\boldsymbol{D})$ is uniformly bounded and $g_t(\boldsymbol{D})$ is Lipschitz. ☐

Proposition 9. *Let $\boldsymbol{D} \in \mathcal{D}$ and denote the minimizer of $\tilde{\ell}(\boldsymbol{z}, \boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e})$ as:*

$$\{\boldsymbol{v}', \boldsymbol{e}'\} = \operatorname*{arg\,min}_{\boldsymbol{v},\boldsymbol{e}} \tilde{\ell}(\boldsymbol{z}, \boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e}).$$

Then, the function $\ell(\boldsymbol{z}, L)$ is continuously differentiable and

$$\nabla_{\boldsymbol{D}} \ell(\boldsymbol{z}, \boldsymbol{D}) = (\boldsymbol{D}\boldsymbol{v}' + \boldsymbol{e}' - \boldsymbol{z})\boldsymbol{v}'^\top.$$

Furthermore, $\ell(\boldsymbol{z}, \cdot)$ is uniformly Lipschitz.

Proof. By fixing the variable $\boldsymbol{z}$, the function $\tilde{\ell}$ can be seen as a mapping:

$$\mathbb{R}^{d+p} \times \mathcal{D} \to \mathbb{R}$$
$$([\boldsymbol{v};\ \boldsymbol{e}], \boldsymbol{D}) \mapsto \tilde{\ell}(\boldsymbol{z}, \boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e}).$$

It is easy to show that for all $[\boldsymbol{v};\ \boldsymbol{e}] \in \mathbb{R}^{d+p}$, $\tilde{\ell}(\boldsymbol{z}, \cdot, \boldsymbol{v}, \boldsymbol{e})$ is differentiable. Also $\tilde{\ell}(\boldsymbol{z}, \cdot, \cdot, \cdot)$ is continuous on $\mathbb{R}^{d+p} \times \mathcal{D}$. $\nabla_{\boldsymbol{D}} \tilde{\ell}(\boldsymbol{z}, \boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e}) = (\boldsymbol{D}\boldsymbol{v} + \boldsymbol{e} - \boldsymbol{z})\boldsymbol{v}^\top$ is continuous on $\mathbb{R}^{d+p} \times \mathcal{D}$. $\forall \boldsymbol{D} \in \mathcal{D}$, since $\tilde{\ell}(\boldsymbol{z}, \boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e})$ is strongly convex w.r.t. $\boldsymbol{v}$, it has a unique minimizer $\{\boldsymbol{v}', \boldsymbol{e}'\}$. Thus Lemma 3 applies and we prove that $\ell(\boldsymbol{z}, \boldsymbol{D})$ is differentiable in $\boldsymbol{D}$ and

$$\nabla_{\boldsymbol{D}} \ell(\boldsymbol{z}, \boldsymbol{D}) = (\boldsymbol{D}\boldsymbol{v}' + \boldsymbol{e}' - \boldsymbol{z})\boldsymbol{v}'^\top.$$

Since every term in $\nabla_{\boldsymbol{D}} \ell(\boldsymbol{z}, \boldsymbol{D})$ is uniformly bounded (Assumption 1 and Proposition 7), we conclude that the gradient of $\ell(\boldsymbol{z}, \boldsymbol{D})$ is uniformly bounded, implying that $\ell(\boldsymbol{z}, \boldsymbol{D})$ is uniformly Lipschitz w.r.t. $\boldsymbol{D}$. ☐

Corollary 10. *Let $f_t(\boldsymbol{D})$ be the empirical loss function defined in (2.10). Then $f_t(\boldsymbol{D})$ is uniformly bounded and Lipschitz.*

Proof. Since $\ell(\boldsymbol{z}, L)$ can be uniformly bounded (Corollary 8), we only need to show that $\frac{1}{t}h(\boldsymbol{Z}, \boldsymbol{D})$ is uniformly bounded. Note that we have derived the form for $h(\boldsymbol{Z}, \boldsymbol{D})$ as follows:

$$\frac{1}{t}h(\boldsymbol{Z}, \boldsymbol{D}) = \frac{1}{t^3} \sum_{i=1}^{t} \frac{1}{2} \left\| \boldsymbol{D}^\top \left(\frac{1}{\lambda_3 t} \boldsymbol{I}_p + \frac{1}{t} \boldsymbol{N}_t \right)^{-1} \boldsymbol{z}_i \right\|_2^2 + \frac{\lambda_3}{2t^3} \left\| \left(\frac{1}{t} \boldsymbol{I}_p + \frac{\lambda_3}{t} \boldsymbol{N}_t \right)^{-1} \boldsymbol{D} \right\|_F^2$$

where $\boldsymbol{N}_t = \sum_{i=1}^{t} \boldsymbol{y}_i \boldsymbol{y}_i^\top$. Since every term in the above equation can be uniformly bounded, $h(\boldsymbol{Z}, \boldsymbol{D})$ is uniformly bounded and so is $f_t(\boldsymbol{D})$.

To show that $f_t(\boldsymbol{D})$ is uniformly Lipschitz, we show that its gradient can be uniformly bounded.

$$\nabla f_t(\boldsymbol{D}) = \frac{1}{t} \sum_{i=1}^{t} \nabla \ell(\boldsymbol{z}_i, \boldsymbol{D}) + \frac{1}{t} \nabla h(\boldsymbol{Z}, \boldsymbol{D})$$
$$= \frac{1}{t} \sum_{i=1}^{t} (\boldsymbol{D}\boldsymbol{v}_i + \boldsymbol{e}_i - \boldsymbol{z}_i)\boldsymbol{v}_i^\top + \frac{1}{t^3} \sum_{i=1}^{t} \left(\frac{1}{\lambda_3 t} \boldsymbol{I}_p + \frac{1}{t} \boldsymbol{N}_t \right)^{-1} \boldsymbol{z}_i \boldsymbol{z}_i^\top \left(\frac{1}{\lambda_3 t} \boldsymbol{I}_p + \frac{1}{t} \boldsymbol{N}_t \right)^{-1} \boldsymbol{D}$$
$$+ \frac{\lambda_3}{t^3} \left(\frac{1}{t} \boldsymbol{I}_p + \frac{\lambda_3}{t} \boldsymbol{N}_t \right)^{-2} \boldsymbol{D}.$$

Then the Frobenius norm of $\nabla f_t(\boldsymbol{D})$ can be bounded by:

$$\begin{aligned}\|\nabla f_t(\boldsymbol{D})\|_F \leq \frac{1}{t}\sum_{i=1}^{t} \|\boldsymbol{D}\boldsymbol{v}_i + \boldsymbol{e}_i - \boldsymbol{z}_i\|_2 \cdot \|\boldsymbol{v}_i\|_2 \\ + \frac{1}{t^3}\sum_{i=1}^{t} \left\| \left(\frac{1}{\lambda_3 t}\boldsymbol{I}_p + \frac{1}{t}\boldsymbol{N}_t \right)^{-1} \right\|_F^2 \cdot \|\boldsymbol{z}_i\|_2^2 \cdot \|\boldsymbol{D}\|_F \\ + \frac{\lambda_3}{t^3} \left\| \left(\frac{1}{t}\boldsymbol{I}_p + \frac{\lambda_3}{t}\boldsymbol{N}_t \right)^{-1} \right\|_F^2 \cdot \|\boldsymbol{D}\|_F \,.\end{aligned}$$

One can easily check that the right side of the inequality is uniformly bounded. Thus $\|\nabla f_t(\boldsymbol{D})\|_F$ is uniformly bounded, implying that $f_t(\boldsymbol{D})$ is uniformly Lipschitz. □

B.2 Proof of P-Donsker

Proposition 11. *Let $f'_t(\boldsymbol{D}) = \frac{1}{t}\sum_{i=1}^{t} \ell(\boldsymbol{z}_i, \boldsymbol{D})$ and $f(\boldsymbol{D})$ be the expected loss function defined in (2.20). Then we have*

$$\mathbb{E}[\sqrt{t}\, \|f'_t - f\|_\infty] = \mathcal{O}(1).$$

Proof. Let us consider $\{\ell(\boldsymbol{z}, \boldsymbol{D})\}$ as a set of measurable functions indexed by $\boldsymbol{D} \in \mathcal{D}$. As we showed in Proposition 7, $\mathcal{D}$ is a compact set. Also, we have proved that $\ell(\boldsymbol{z}, \boldsymbol{D})$ is uniformly Lipschitz over $\boldsymbol{D}$ (Proposition 9). Thus, $\{\ell(\boldsymbol{z}, \boldsymbol{D})\}$ is P-Donsker (see the definition in Lemma 4). Furthermore, as $\ell(\boldsymbol{z}, \boldsymbol{D})$ is non-negative and uniformly bounded, so is $\ell^2(\boldsymbol{z}, \boldsymbol{D})$. So we have $\mathbb{E}_{\boldsymbol{z}}[\ell^2(\boldsymbol{z}, \boldsymbol{D})]$ being uniformly bounded. Since we have verified all the hypotheses in Lemma 4, we obtain the result that

$$\mathbb{E}[\sqrt{t}\, \|f'_t - f\|_\infty] = \mathcal{O}(1).$$

□

B.3 Proof of convergence of $g_t(D)$

Theorem 12 (Convergence of the surrogate function $g_t(\boldsymbol{D}_t)$)**.** *The surrogate function $g_t(\boldsymbol{D}_t)$ we defined in (3.6) converges almost surely, where $\boldsymbol{D}_t$ is the solution produced by Algorithm 1.*

Proof. Note that $g_t(\boldsymbol{D}_t)$ can be viewed as a stochastic positive process since every term in $g_t(\boldsymbol{D}_t)$ is non-negative and the samples are drawn randomly. We define

$$u_t = g_t(\boldsymbol{D}_t).$$

To show the convergence of u_t, we need to bound the difference of u_{t+1} and u_t:

$$
\begin{aligned}
& u_{t+1} - u_t \\
&= g_{t+1}(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_t) \\
&= g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t) + g_{t+1}(\boldsymbol{D}_t) - g_t(\boldsymbol{D}_t) \\
&= g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t) + \frac{1}{t+1}\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) - \frac{1}{t+1} g'_t(\boldsymbol{D}_t) \\
&\quad + \left[\frac{1}{t+1} \sum_{i=1}^{t+1} \frac{1}{2} \|\boldsymbol{u}_i\|_2^2 + \frac{\lambda_3}{2(t+1)} \|\boldsymbol{D}_t - \boldsymbol{M}_{t+1}\|_F^2 - \frac{1}{t} \sum_{i=1}^{t} \frac{1}{2} \|\boldsymbol{u}_i\|_2^2 - \frac{\lambda_3}{2t} \|\boldsymbol{D}_t - \boldsymbol{M}_t\|_F^2 \right] \\
&= g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t) + \frac{f'_t(\boldsymbol{D}_t) - g'_t(\boldsymbol{D}_t)}{t+1} + \frac{\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) - f'_t(\boldsymbol{D}_t)}{t+1} \\
&\quad + \left[\frac{1}{t+1} \sum_{i=1}^{t+1} \frac{1}{2} \|\boldsymbol{u}_i\|_2^2 + \frac{\lambda_3}{2(t+1)} \|\boldsymbol{D}_t - \boldsymbol{M}_{t+1}\|_F^2 - \frac{1}{t} \sum_{i=1}^{t} \frac{1}{2} \|\boldsymbol{u}_i\|_2^2 - \frac{\lambda_3}{2t} \|\boldsymbol{D}_t - \boldsymbol{M}_t\|_F^2 \right].
\end{aligned} \tag{B.12}
$$

Here,

$$
g'_t(\boldsymbol{D}_t) = \frac{1}{t} \sum_{i=1}^{t} \tilde{\ell}(\boldsymbol{z}_i, \boldsymbol{D}, \boldsymbol{v}_i, \boldsymbol{e}_i). \tag{B.13}
$$

First, we bound the last four terms. We have

$$
\frac{1}{t+1} \sum_{i=1}^{t+1} \frac{1}{2} \|\boldsymbol{u}_i\|_2^2 - \frac{1}{t} \sum_{i=1}^{t} \|\boldsymbol{u}_i\|_2^2 = \frac{-1}{t(t+1)} \sum_{i=1}^{t} \frac{1}{2} \|\boldsymbol{u}_i\|_2^2 + \frac{1}{2(t+1)} \|\boldsymbol{u}_{t+1}\|_2^2 \leq \frac{1}{2(t+1)} \|\boldsymbol{u}_{t+1}\|_2^2 . \tag{B.14}
$$

Also we have

$$
\begin{aligned}
& \frac{\lambda_3}{2(t+1)} \|\boldsymbol{D}_t - \boldsymbol{M}_{t+1}\|_F^2 - \frac{\lambda_3}{2t} \|\boldsymbol{D}_t - \boldsymbol{M}_t\|_F^2 \\
&= \frac{-\lambda_3}{2t(t+1)} \|\boldsymbol{D}_t - \boldsymbol{M}_t\|_F^2 + \frac{\lambda_3}{2(t+1)} \left\| \boldsymbol{z}_{t+1} \boldsymbol{u}_{t+1}^\top \right\|_F^2 - \frac{\lambda_3}{t+1} \operatorname{Tr}\left((\boldsymbol{D}_t - \boldsymbol{M}_t)^\top \boldsymbol{z}_{t+1} \boldsymbol{u}_{t+1}^\top \right) \\
&= \frac{-\lambda_3}{2t(t+1)} \|\boldsymbol{D}_t - \boldsymbol{M}_t\|_F^2 + \frac{\lambda_3}{2(t+1)} \left\| \boldsymbol{z}_{t+1} \boldsymbol{u}_{t+1}^\top \right\|_F^2 - \frac{\lambda_3}{t+1} \left(\|\boldsymbol{z}_{t+1}\|_2^2 + \frac{1}{\lambda_3} \right) \|\boldsymbol{u}_{t+1}\|_2^2 \\
&\leq \frac{1}{t+1} \left(\frac{\lambda_3}{2} \left\| \boldsymbol{z}_{t+1} \boldsymbol{u}_{t+1}^\top \right\|_F^2 - (\lambda_3 \|\boldsymbol{z}_{t+1}\|_2^2 + 1) \|\boldsymbol{u}_{t+1}\|_2^2 \right) \\
&\leq \frac{1}{t+1} \left(-\frac{\lambda_3}{2} \|\boldsymbol{z}_{t+1}\|_2^2 \|\boldsymbol{u}_{t+1}\|_2^2 - \|\boldsymbol{u}_{t+1}\|_2^2 \right),
\end{aligned} \tag{B.15}
$$

where the first equality is derived by the fact that $\boldsymbol{M}_{t+1} = \boldsymbol{M}_t + \boldsymbol{z}_{t+1} \boldsymbol{u}_{t+1}^\top$, and the second equality is derived by the closed form solution of $\boldsymbol{u}_{t+1}$ (see (3.5)).

Combining (B.14) and (B.15), we know that

$$
\begin{aligned}
& \frac{1}{t+1} \sum_{i=1}^{t+1} \frac{1}{2} \|\boldsymbol{u}_i\|_2^2 - \frac{1}{t} \sum_{i=1}^{t} \|\boldsymbol{u}_i\|_2^2 + \frac{\lambda_3}{2(t+1)} \|\boldsymbol{D}_t - \boldsymbol{M}_{t+1}\|_F^2 - \frac{\lambda_3}{2t} \|\boldsymbol{D}_t - \boldsymbol{M}_t\|_F^2 \\
&\leq \frac{1}{2(t+1)} \|\boldsymbol{u}_{t+1}\|_2^2 + \frac{1}{t+1} \Big(-\frac{\lambda_3}{2} \|\boldsymbol{z}_{t+1}\|_2^2 \|\boldsymbol{u}_{t+1}\|_2^2 - \|\boldsymbol{u}_{t+1}\|_2^2 \Big) \\
&= \frac{1}{t+1} \left(-\frac{\lambda_3}{2} \|\boldsymbol{z}_{t+1}\|_2^2 \|\boldsymbol{u}_{t+1}\|_2^2 - \frac{1}{2} \|\boldsymbol{u}_{t+1}\|_2^2 \right) \leq 0.
\end{aligned}
$$

Therefore,

$$\begin{aligned}
u_{t+1} - u_t &\le g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t) + \frac{1}{t+1}\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) - \frac{1}{t+1} g'_t(\boldsymbol{D}_t) \\
&= g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t) + \frac{f'_t(\boldsymbol{D}_t) - g'_t(\boldsymbol{D}_t)}{t+1} + \frac{\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) - f'_t(\boldsymbol{D}_t)}{t+1} \\
&\le \frac{\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) - f'_t(\boldsymbol{D}_t)}{t+1},
\end{aligned}$$

where $f'_t(\boldsymbol{D})$ is defined in Proposition 11, and the last inequality holds because $\boldsymbol{D}_{t+1}$ is the minimizer of $g_{t+1}(\boldsymbol{D})$ and $g'_t(\boldsymbol{D})$ is a surrogate function of $f'_t(\boldsymbol{D})$.

Let $\mathcal{F}_t$ be the filtration of the past information. We take the expectation on the above equation conditioned on $\mathcal{F}_t$:

$$\begin{aligned}
\mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t] &\le \frac{\mathbb{E}[\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) \mid \mathcal{F}_t] - f'_t(\boldsymbol{D}_t)}{t+1} \\
&\le \frac{f(\boldsymbol{D}_t) - f'_t(\boldsymbol{D}_t)}{t+1} \\
&\le \frac{\|f - f'_t\|_\infty}{t+1}.
\end{aligned}$$

From Proposition 11, we know

$$\mathbb{E}[\left\|f - f'_t\right\|_\infty] = \mathcal{O}(\frac{1}{\sqrt{t}}).$$

Thus,

$$\mathbb{E}\big[\mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t]^+\big] = \mathbb{E}\big[\max\{\mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t], 0\}\big] \le \frac{c}{\sqrt{t}(t+1)},$$

where c is some constant.

Now let us define the index set

$$\mathcal{T} = \{t \mid \mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t] > 0\},$$

and the indicator

$$\delta_t = \begin{cases} 1, & \text{if } t \in \mathcal{T}, \\ 0, & \text{otherwise}. \end{cases}$$

We have

$$\begin{aligned}
\sum_{t=1}^{\infty} \mathbb{E}\big[\delta_t(u_{t+1} - u_t)\big] &= \sum_{t \in \mathcal{T}} \mathbb{E}[u_{t+1} - u_t] \\
&= \sum_{t \in \mathcal{T}} \mathbb{E}\big[\mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t]\big] \\
&= \sum_{t=1}^{\infty} \mathbb{E}\big[\mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t]^+\big] \\
&\le +\infty.
\end{aligned}$$

Thus, Lemma 5 applies. That is, $g_t(\boldsymbol{D}_t)$ is a quasi-martingale and converges almost surely. Moreover,

$$\sum_{t=1}^{\infty} |\mathbb{E}[u_{t+1} - u_t \mid \mathcal{F}_t]| < +\infty, \ a.s. \tag{B.16}$$

□

B.4 Proof of Convergence of $\boldsymbol{D}_t$

Proposition 13. *Let $\{\boldsymbol{D}_t\}_{t=1}^{\infty}$ be the basis sequence produced by the Algorithm 1. Then,*

$$\left\|\boldsymbol{D}_{t+1}-\boldsymbol{D}_t\right\|_F=\mathcal{O}\left(\frac{1}{t}\right). \tag{B.17}$$

Proof. According the strong convexity of $g_t(\boldsymbol{D})$ (Assumption 3), we have,

$$g_t(\boldsymbol{D}_{t+1})-g_t(\boldsymbol{D}_t)\geq\frac{\beta_0}{2}\left\|\boldsymbol{D}_{t+1}-\boldsymbol{D}_t\right\|_F^2, \tag{B.18}$$

On the other hand,

$$\begin{aligned}
&g_t(\boldsymbol{D}_{t+1})-g_t(\boldsymbol{D}_t)\\
=\,&g_t(\boldsymbol{D}_{t+1})-g_{t+1}(\boldsymbol{D}_{t+1})+g_{t+1}(\boldsymbol{D}_{t+1})-g_{t+1}(\boldsymbol{D}_t)+g_{t+1}(\boldsymbol{D}_t)-g_t(\boldsymbol{D}_t)\\
\leq\,&g_t(\boldsymbol{D}_{t+1})-g_{t+1}(\boldsymbol{D}_{t+1})+g_{t+1}(\boldsymbol{D}_t)-g_t(\boldsymbol{D}_t)\\
\overset{\text{def}}{=}\,&G_t(\boldsymbol{D}_{t+1})-G_t(\boldsymbol{D}_t).
\end{aligned} \tag{B.19}$$

Note that the inequality is derived by the fact that $g_{t+1}(\boldsymbol{D}_{t+1})-g_{t+1}(\boldsymbol{D}_t)\leq 0$, as $\boldsymbol{D}_{t+1}$ is the minimizer of $g_{t+1}(\boldsymbol{D})$. We denote $g_t(\boldsymbol{D})-g_{t+1}(\boldsymbol{D})$ by $G_t(\boldsymbol{D})$.

By a simple calculation, we obtain the gradient of $G_t(\boldsymbol{D})$:

$$\begin{aligned}
\nabla G_t(\boldsymbol{D})&=\nabla g_t(\boldsymbol{D})-\nabla g_{t+1}(\boldsymbol{D})\\
&=\frac{1}{t}\Big[\boldsymbol{D}\left(\lambda_1\boldsymbol{A}_t+\lambda_3\boldsymbol{I}_d\right)-\left(\lambda_1\boldsymbol{B}_t+\lambda_3\boldsymbol{M}_t\right)\Big]\\
&\quad-\frac{1}{t+1}\Big[\boldsymbol{D}(\lambda_1A_{t+1}+\lambda_3\boldsymbol{I}_d)-(\lambda_1B_{t+1}+\lambda_3\boldsymbol{M}_{t+1})\Big]\\
&=\frac{1}{t}\Bigg[\boldsymbol{D}\left(\lambda_1\boldsymbol{A}_t+\lambda_3\boldsymbol{I}_d-\frac{\lambda_1t}{t+1}A_{t+1}-\frac{\lambda_3t}{t+1}\boldsymbol{I}_d\right)\\
&\quad+\frac{\lambda_1t}{t+1}B_{t+1}-\lambda_1\boldsymbol{B}_t+\frac{\lambda_3t}{t+1}\boldsymbol{M}_{t+1}-\lambda_3\boldsymbol{M}_t\Bigg]\\
&=\frac{1}{t}\Bigg[\boldsymbol{D}\left(\frac{\lambda_1}{t+1}A_{t+1}-\lambda_1\boldsymbol{v}_{t+1}\boldsymbol{v}_{t+1}^\top+\frac{\lambda_3}{t+1}\boldsymbol{I}_d\right)\\
&\quad+\lambda_1(\boldsymbol{z}_{t+1}-\boldsymbol{e}_{t+1})\boldsymbol{v}_{t+1}^\top-\frac{\lambda_1}{t+1}B_{t+1}+\lambda_3\boldsymbol{z}_{t+1}\boldsymbol{u}_{t+1}^\top-\frac{\lambda_3}{t+1}\boldsymbol{M}_{t+1}\Bigg]
\end{aligned}$$

So the Frobenius norm of $\nabla G_t(\boldsymbol{D})$ follows immediately:

$$
\begin{aligned}
&\left\|\nabla G_t(\boldsymbol{D})\right\|_F \\
\leq&\frac{1}{t}\Bigg[\left\|\boldsymbol{D}\right\|_F\left(\lambda_1\left\|\frac{A_{t+1}}{t+1}\right\|_F+\lambda_1\left\|\boldsymbol{v}_{t+1}\boldsymbol{v}_{t+1}^\top\right\|_F+\frac{\lambda_3}{t+1}\left\|\boldsymbol{I}_d\right\|_F\right)+\lambda_1\left\|(\boldsymbol{z}_{t+1}-\boldsymbol{e}_{t+1})\boldsymbol{v}_{t+1}^\top\right\|_F \\
&+\lambda_1\left\|\frac{B_{t+1}}{t+1}\right\|_F+\lambda_3\left\|\boldsymbol{z}_{t+1}\boldsymbol{u}_{t+1}^\top\right\|_F+\frac{\lambda_3}{t+1}\left\|\boldsymbol{M}_{t+1}\right\|_F\Bigg] \\
=&\frac{1}{t}\Bigg[\left\|\boldsymbol{D}\right\|_F\left(\lambda_1\left\|\frac{A_{t+1}}{t+1}\right\|_F+\lambda_1\left\|\boldsymbol{v}_{t+1}\boldsymbol{v}_{t+1}^\top\right\|_F\right)+\lambda_1\left\|(\boldsymbol{z}_{t+1}-\boldsymbol{e}_{t+1})\boldsymbol{v}_{t+1}^\top\right\|_F \\
&+\lambda_1\left\|\frac{B_{t+1}}{t+1}\right\|_F+\lambda_3\left\|\boldsymbol{z}_{t+1}\boldsymbol{u}_{t+1}^\top\right\|_F\Bigg]+\frac{\lambda_3}{t(t+1)}\left(\left\|\boldsymbol{I}_d\right\|_F+\left\|\boldsymbol{M}_{t+1}\right\|_F\right).
\end{aligned}
$$

We know from Proposition 7 that all the terms in the above equation are uniformly bounded. Thus, there exist constants c_1, c_2 and c_3, such that

$$
\left\|\nabla G_t(\boldsymbol{D})\right\|_F \leq \frac{1}{t}\left(c_1\left\|\boldsymbol{D}\right\|_F+c_2\right)+\frac{c_3}{t(t+1)}.
$$

According to the first order Taylor expansion,

$$
\begin{aligned}
G_t(\boldsymbol{D}_{t+1})-G_t(\boldsymbol{D}_t)=&\ \operatorname{Tr}\left(\left(\boldsymbol{D}_{t+1}-\boldsymbol{D}_t\right)^\top\nabla G_t\left(\alpha\boldsymbol{D}_t+(1-\alpha)\,\boldsymbol{D}_{t+1}\right)\right) \\
\leq&\ \left\|\boldsymbol{D}_{t+1}-\boldsymbol{D}_t\right\|_F\cdot\left\|\nabla G_t\left(\alpha\boldsymbol{D}_t+(1-\alpha)\,\boldsymbol{D}_{t+1}\right)\right\|_F,
\end{aligned}
$$

where α is a constant between 0 and 1. According to Proposition 7, $\boldsymbol{D}_t$ and $\boldsymbol{D}_{t+1}$ are uniformly bounded, so $\alpha\boldsymbol{D}_t+(1-\alpha)\,\boldsymbol{D}_{t+1}$ is uniformly bounded. Thus, there exists a constant c_4, such that

$$
\left\|\nabla G_t\left(\alpha L_t+(1-\alpha)\,L_{t+1}\right)\right\|_F \leq \frac{c_4}{t}+\frac{c_3}{t(t+1)},
$$

resulting in

$$
G_t(\boldsymbol{D}_{t+1})-G_t(\boldsymbol{D}_t) \leq \left(\frac{c_4}{t}+\frac{c_3}{t(t+1)}\right)\left\|\boldsymbol{D}_{t+1}-\boldsymbol{D}_t\right\|_F.
$$

Combining (B.18), (B.19) and the above equation, we have

$$
\left\|\boldsymbol{D}_{t+1}-\boldsymbol{D}_t\right\|_F=\frac{2c_4}{\beta_0}\cdot\frac{1}{t}+\frac{2c_3}{\beta_0}\cdot\frac{1}{t(t+1)}.
$$

□

B.5 Proof for convergence of $f_t(\boldsymbol{D}_t)$

Theorem 14 (Convergence of $f_t(\boldsymbol{D}_t)$). *Let $f_t(\boldsymbol{D}_t)$ be the empirical loss function defined in (2.10) and $\boldsymbol{D}_t$ be the solution produced by the Algorithm 1. Let $b_t = g_t(\boldsymbol{D}_t) - f_t(\boldsymbol{D}_t)$. Then, b_t converges almost surely to 0. Thus, $f_t(\boldsymbol{D}_t)$ converges almost surely to the same limit of $g_t(\boldsymbol{D}_t)$.*

Proof. Let $f_t'(\boldsymbol{D})$ and $g_t'(\boldsymbol{D})$ be those defined in Proposition 11 and Theorem 12 respectively. Then,

$$
\begin{aligned}
b_t &= g_t(\boldsymbol{D}_t) - f_t(\boldsymbol{D}_t) \\
&= g_t'(\boldsymbol{D}_t) - f_t'(\boldsymbol{D}_t) + \left[\frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i\right\|_2^2 + \frac{\lambda_3}{2t}\left\|\boldsymbol{D}_t - \boldsymbol{M}_t\right\|_F^2\right. \\
&\quad \left. - \frac{1}{t^3}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{D}_t^\top\left(\frac{1}{\lambda_3 t}\boldsymbol{I}_p + \frac{1}{t}\boldsymbol{N}_t\right)^{-1}\boldsymbol{z}_i\right\|_2^2 - \frac{\lambda_3}{2t^3}\left\|\left(\frac{1}{t}\boldsymbol{I}_p + \frac{\lambda_3}{t}\boldsymbol{N}_t\right)^{-1}\boldsymbol{D}_t\right\|_F^2\right] \\
&= g_t'(\boldsymbol{D}_t) - f_t'(\boldsymbol{D}_t) + q_t(\boldsymbol{D}_t),
\end{aligned}
$$

where $q_t(\boldsymbol{D}_t)$ denotes the last four terms. Combining B.12, we have

$$
\begin{aligned}
&\frac{b_t}{t+1} \\
&= \frac{g_t'(\boldsymbol{D}_t) - f_t'(\boldsymbol{D}_t)}{t+1} + \frac{q_t(\boldsymbol{D}_t)}{t+1} \\
&= g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t) + \frac{\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) - f_t'(\boldsymbol{D}_t)}{t+1} + u_t - u_{t+1} \\
&\quad + \left[\frac{q_t(\boldsymbol{D}_t)}{t+1} + \frac{1}{t+1}\sum_{i=1}^{t+1}\frac{1}{2}\left\|\boldsymbol{u}_i\right\|_2^2 + \frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{D}_t - \boldsymbol{M}_{t+1}\right\|_F^2 - \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i\right\|_2^2 - \frac{\lambda_3}{2t}\left\|\boldsymbol{D}_t - \boldsymbol{M}_t\right\|_F^2\right].
\end{aligned}
$$

Note that we naturally have

$$
\begin{aligned}
\frac{q_t(\boldsymbol{D}_t)}{t+1} &\le \frac{1}{t(t+1)}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i\right\|_2^2 + \frac{\lambda_3}{2t(t+1)}\left\|\boldsymbol{D}_t - \boldsymbol{M}_t\right\|_F^2 \\
&\le \frac{1}{t(t+1)}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i\right\|_2^2 + \frac{c}{2t(t+1)},
\end{aligned}
$$

where the second inequality holds as $\boldsymbol{D}_t$ and $\boldsymbol{M}_t$ are both uniformly bounded (see Proposition 7).

Also, from (B.14), we know

$$
\frac{1}{t+1}\sum_{i=1}^{t+1}\frac{1}{2}\left\|\boldsymbol{u}_i\right\|_2^2 - \frac{1}{t}\sum_{i=1}^{t}\left\|\boldsymbol{u}_i\right\|_2^2 = \frac{-1}{t(t+1)}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i\right\|_2^2 + \frac{1}{2(t+1)}\left\|\boldsymbol{u}_{t+1}\right\|_2^2.
$$

And from (B.15)

$$
\frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{D}_t - \boldsymbol{M}_{t+1}\right\|_F^2 - \frac{\lambda_3}{2t}\left\|\boldsymbol{D}_t - \boldsymbol{M}_t\right\|_F^2 \le \frac{1}{t+1}\left(-\frac{\lambda_3}{2}\left\|\boldsymbol{z}_{t+1}\right\|_2^2\left\|\boldsymbol{u}_{t+1}\right\|_2^2 - \left\|\boldsymbol{u}_{t+1}\right\|_2^2\right).
$$

Thus,

$$
\begin{aligned}
&\frac{q_t(\boldsymbol{D}_t)}{t+1} + \frac{1}{t+1}\sum_{i=1}^{t+1}\frac{1}{2}\left\|\boldsymbol{u}_i\right\|_2^2 + \frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{D}_t - \boldsymbol{M}_{t+1}\right\|_F^2 - \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\left\|\boldsymbol{u}_i\right\|_2^2 - \frac{\lambda_3}{2t}\left\|\boldsymbol{D}_t - \boldsymbol{M}_t\right\|_F^2 \\
&\le \frac{c}{2t(t+1)} + \frac{1}{2(t+1)}\left\|\boldsymbol{u}_{t+1}\right\|_2^2 + \frac{1}{t+1}\left(-\frac{\lambda_3}{2}\left\|\boldsymbol{z}_{t+1}\right\|_2^2\left\|\boldsymbol{u}_{t+1}\right\|_2^2 - \left\|\boldsymbol{u}_{t+1}\right\|_2^2\right) \\
&= \frac{c}{2t(t+1)} - \frac{1}{2(t+1)}\left\|\boldsymbol{u}_{t+1}\right\|_2^2 - \frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{z}_{t+1}\right\|_2^2\left\|\boldsymbol{u}_{t+1}\right\|_2^2 \\
&\le \frac{c}{2t(t+1)}.
\end{aligned}
$$

Therefore,

$$\begin{aligned}\frac{b_t}{t+1} &\le g_{t+1}(\boldsymbol{D}_{t+1}) - g_{t+1}(\boldsymbol{D}_t) + \frac{\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) - f_t'(\boldsymbol{D}_t)}{t+1} + u_t - u_{t+1} + \frac{c}{2t(t+1)} \\ &\le \frac{\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_t) - f_t'(\boldsymbol{D}_t)}{t+1} + u_t - u_{t+1} + \frac{c}{2t(t+1)}.\end{aligned}$$

By taking the expectation conditioned on the past information $\mathcal{F}_t$, we have

$$\begin{aligned}\frac{b_t}{t+1} &\le \frac{f(\boldsymbol{D}_t) - f_t(\boldsymbol{D}_t)}{t+1} + \mathbb{E}[u_t - u_{t+1} \mid \mathcal{F}_t] + \frac{c}{2t(t+1)} \\ &\le \frac{c_1}{\sqrt{t}(t+1)} + |\mathbb{E}[u_t - u_{t+1} \mid \mathcal{F}_t]| + \frac{c}{2t(t+1)},\end{aligned}$$

where the second inequality holds by applying Proposition 11. Thus,

$$\sum_{t=1}^{\infty} \frac{b_t}{t+1} \le \sum_{t=1}^{\infty} \frac{c_1}{\sqrt{t}(t+1)} + \sum_{t=1}^{\infty} |\mathbb{E}[u_t - u_{t+1} \mid \mathcal{F}_t]| + \sum_{t=1}^{\infty} \frac{c}{2t(t+1)} < +\infty.$$

Here, the last inequality is derived by applying (B.16).

Next, we examine the difference between b_{t+1} and b_t:

$$\begin{aligned}|b_{t+1} - b_t| =\ &|g_{t+1}(\boldsymbol{D}_{t+1}) - f_{t+1}(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_t) + f_t(\boldsymbol{D}_t)| \\ \le\ &|g_{t+1}(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_{t+1})| + |g_t(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_t)| \\ &+ |f_{t+1}(\boldsymbol{D}_{t+1}) - f_t(\boldsymbol{D}_{t+1})| + |f_t(\boldsymbol{D}_{t+1}) - f_t(\boldsymbol{D}_t)|.\end{aligned} \tag{B.20}$$

For the first term on the right hand side,

$$\begin{aligned}&|g_{t+1}(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_{t+1})| \\ =\ &\Big| g_{t+1}'(\boldsymbol{D}_{t+1}) - g_t'(\boldsymbol{D}_{t+1}) + \frac{1}{t+1}\sum_{i=1}^{t+1} \frac{1}{2}\|\boldsymbol{u}_i\|_2^2 \\ &- \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\|\boldsymbol{u}_i\|_2^2 + \frac{\lambda_3}{2(t+1)}\|\boldsymbol{D}_{t+1} - \boldsymbol{M}_{t+1}\|_F^2 - \frac{\lambda_3}{2t}\|\boldsymbol{D}_{t+1} - \boldsymbol{M}_t\|_F^2 \Big| \\ =\ &\Big| g_{t+1}'(\boldsymbol{D}_{t+1}) - g_t'(\boldsymbol{D}_{t+1}) - \frac{1}{t(t+1)}\sum_{i=1}^{t}\frac{1}{2}\|\boldsymbol{u}_i\|_2^2 - \frac{1}{2(t+1)}\|\boldsymbol{u}_{t+1}\|_2^2 \\ &- \frac{\lambda_3}{2t(t+1)}\|\boldsymbol{D}_{t+1} - \boldsymbol{M}_t\|_F^2 - \frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{z}_{t+1}\boldsymbol{u}_{t+1}^\top\right\|_F^2 \Big| \\ \le\ &\left|g_{t+1}'(\boldsymbol{D}_{t+1}) - g_t'(\boldsymbol{D}_{t+1})\right| + \frac{1}{t(t+1)}\sum_{i=1}^{t}\frac{1}{2}\|\boldsymbol{u}_i\|_2^2 \\ &+ \frac{1}{2(t+1)}\|\boldsymbol{u}_{t+1}\|_2^2 + \frac{\lambda_3}{2t(t+1)}\|\boldsymbol{D}_{t+1} - \boldsymbol{M}_t\|_F^2 + \frac{\lambda_3}{2(t+1)}\left\|\boldsymbol{z}_{t+1}\boldsymbol{u}_{t+1}^\top\right\|_F^2 \\ \overset{\zeta_1}{\le}\ &\left|g_{t+1}'(\boldsymbol{D}_{t+1}) - g_t'(\boldsymbol{D}_{t+1})\right| + \frac{c_1}{t+1} \\ =\ &\left|\frac{1}{t+1}\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_{t+1}) - \frac{1}{t+1}g_t'(\boldsymbol{D}_{t+1})\right| + \frac{c_1}{t+1} \\ \overset{\zeta_2}{\le}\ &\frac{c_2}{t+1},\end{aligned}$$

where c_1 and c_2 are some uniform constants. Note that ζ_1 holds because all the $\boldsymbol{u}_i$'s, $\boldsymbol{D}_{t+1}$, $\boldsymbol{M}_t$ and $\boldsymbol{z}_{t+1}$ are uniformly bounded (see Proposition 7), and ζ_2 holds because $\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_{t+1})$ and $g_t'(\boldsymbol{D}_{t+1})$ are uniformly bounded (see Corollary 8).

For the third term on the right hand side of (B.20), we can similarly derive

$$
\begin{aligned}
|f_{t+1}(\boldsymbol{D}_{t+1}) - f_t(\boldsymbol{D}_{t+1})| &\leq \left|f_{t+1}'(\boldsymbol{D}_{t+1}) - f_t'(\boldsymbol{D}_{t+1})\right| + \frac{c_3}{t+1} \\
&= \left|\frac{1}{t+1}\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_{t+1}) - \frac{1}{t+1} f_t'(\boldsymbol{D}_{t+1})\right| + \frac{c_3}{t+1} \\
&\overset{\zeta_3}{\leq} \frac{c_4}{t+1},
\end{aligned}
$$

where c_3 and c_4 are some uniform constants, and ζ_3 holds as $\ell(\boldsymbol{z}_{t+1}, \boldsymbol{D}_{t+1})$ and $f_t'(\boldsymbol{D}_{t+1})$ are both uniformly bounded (see Corollary 10).

From Corollary 8 and Corollary 10, we know that both $g_t(\boldsymbol{D})$ and $f_t(\boldsymbol{D})$ are uniformly Lipschitz. That is, there exists uniform constants κ_1, κ_2, such that

$$
\begin{aligned}
|g_t(\boldsymbol{D}_{t+1}) - g_t(\boldsymbol{D}_t)| \leq& \kappa_1 \left\|\boldsymbol{D}_{t+1} - \boldsymbol{D}_t\right\|_F \overset{\zeta_4}{\leq} \frac{\kappa_3}{t+1}, \\
|f_t(\boldsymbol{D}_{t+1}) - f_t(\boldsymbol{D}_t)| \leq& \kappa_2 \left\|\boldsymbol{D}_{t+1} - \boldsymbol{D}_t\right\|_F \overset{\zeta_5}{\leq} \frac{\kappa_4}{t+1}.
\end{aligned}
$$

Here, ζ_4 and ζ_5 are derived by applying Proposition 13 and κ_3 and κ_4 are some uniform constants.

Finally, we have a bound for (B.20):

$$
|b_{t+1} - b_t| \leq \frac{\kappa_0}{t+1},
$$

where κ_0 is some uniform constant.

By applying Lemma 6, we conclude that $\{b_t\}$ converges to zero. That is,

$$
\lim_{t \to +\infty} g_t(\boldsymbol{D}_t) - f_t(\boldsymbol{D}_t) = 0.
$$

Since we have proved in Theorem 12 that $g_t(\boldsymbol{D}_t)$ converges almost surely, we conclude that $f_t(\boldsymbol{D}_t)$ converges almost surely to the same limit of $g_t(\boldsymbol{D}_t)$. □

Theorem 15 (Convergence of $f(\boldsymbol{D}_t)$)**.** *Let $f(\boldsymbol{D})$ be the expected loss function we defined in* (2.20) *and let $\boldsymbol{D}_t$ be the optimal solution produced by Algorithm 1. Then $f(\boldsymbol{D}_t)$ converges almost surely to the same limit of $f_t(\boldsymbol{D}_t)$ (or, $g_t(\boldsymbol{D}_t)$).*

Proof. According to the central limit theorem, we know that $\sqrt{t}(f(\boldsymbol{D}_t) - f_t(\boldsymbol{D}_t))$ is bounded, implying

$$
\lim_{t \to +\infty} f(\boldsymbol{D}_t) - f_t(\boldsymbol{D}_t) = 0, \quad a.s.
$$

□

B.6 Proof of gradient of $f(\boldsymbol{D})$

Proposition 16 (Gradient of $f(\boldsymbol{D})$)**.** *Let $f(\boldsymbol{D})$ be the expected loss function which is defined in* (2.20)*. Then, $f(\boldsymbol{D})$ is continuously differentiable and $\nabla f(\boldsymbol{D}) = \mathbb{E}_{\boldsymbol{z}}[\nabla_{\boldsymbol{D}} \ell(\boldsymbol{z}, \boldsymbol{D})]$. Moreover, $\nabla f(\boldsymbol{D})$ is uniformly Lipschitz on $\mathcal{D}$.*

Proof. We have shown in Proposition 9 that $\ell(\boldsymbol{z}, \boldsymbol{D})$ is continuously differentiable, $f(\boldsymbol{D})$ is also continuously differentiable and we have $\nabla f(\boldsymbol{D}) = \mathbb{E}_{\boldsymbol{z}}[\nabla_{\boldsymbol{D}} \ell(\boldsymbol{z}, \boldsymbol{D})]$.

Next, we prove the Lipschitz of $\nabla f(\boldsymbol{D})$. Let $\boldsymbol{v}'(\boldsymbol{z}', D')$ and $\boldsymbol{e}'(\boldsymbol{z}', \boldsymbol{D}')$ be the minimizer of $\tilde{\ell}(\boldsymbol{z}', \boldsymbol{D}', \boldsymbol{v}, \boldsymbol{e})$. Since $\tilde{\ell}(\boldsymbol{z}, \boldsymbol{D}, \boldsymbol{v}, \boldsymbol{e})$ has a unique minimum and is continuous in $\boldsymbol{z}$, $\boldsymbol{D}$, $\boldsymbol{v}$ and $\boldsymbol{e}$, $\boldsymbol{v}'(\boldsymbol{z}', \boldsymbol{D}')$ and $\boldsymbol{e}'(\boldsymbol{z}', \boldsymbol{D}')$ is continuous in $\boldsymbol{z}$ and $\boldsymbol{D}$.

Let $\Lambda = \{j \mid \boldsymbol{e}'_j \neq 0\}$. According the first order optimality condition, we know that

$$\frac{\partial \tilde{\ell}(\boldsymbol{z}, D, \boldsymbol{v}, \boldsymbol{e})}{\partial \boldsymbol{e}} = 0,$$

which implies

$$\lambda_1(\boldsymbol{z} - D\boldsymbol{v} - \boldsymbol{e}) \in \lambda_2 \left\| \boldsymbol{e} \right\|_1.$$

Hence,

$$|(\boldsymbol{z} - \boldsymbol{D}\boldsymbol{v} - \boldsymbol{e})_j| = \frac{\lambda_2}{\lambda_1}, \ \forall j \in \Lambda.$$

Since $\boldsymbol{z} - \boldsymbol{D}\boldsymbol{v} - \boldsymbol{e}$ is continuous in $\boldsymbol{z}$ and $\boldsymbol{D}$, there exists an open neighborhood $\mathcal{V}$, such that for all $(\boldsymbol{z}'', \boldsymbol{D}'') \in \mathcal{V}$, if $j \notin \Lambda$, then $|(\boldsymbol{z}'' - \boldsymbol{D}''\boldsymbol{v}'' - \boldsymbol{e}'')_j| < \frac{\lambda_2}{\lambda_1}$ and $\boldsymbol{e}''_j = 0$. That is, the support set of $\boldsymbol{e}'$ will not change.

Let us denote $\boldsymbol{H} = [\boldsymbol{D} \ \boldsymbol{I}_p]$, $\boldsymbol{r} = [\boldsymbol{v}^\top \ \boldsymbol{e}^\top]^\top$ and define the function

$$\tilde{\ell}(\boldsymbol{z}, \boldsymbol{H}, \boldsymbol{r}_\Lambda) = \frac{\lambda_1}{2} \left\| \boldsymbol{z} - \boldsymbol{H}_\Lambda \boldsymbol{r}_\Lambda \right\|_2^2 + \frac{1}{2} \left\| [\boldsymbol{I} \ 0] \boldsymbol{r}_\Lambda \right\|_2^2 + \lambda_2 \left\| [0 \ \boldsymbol{I}] \boldsymbol{r}_\Lambda \right\|_1.$$

Since $\tilde{\ell}(\boldsymbol{z}, \boldsymbol{D}_\Lambda, \cdot)$ is strongly convex, there exists a uniform constant κ_1, such that for all $\boldsymbol{r}''_\Lambda$,

$$\tilde{\ell}(\boldsymbol{z}', \boldsymbol{H}'_\Lambda, \boldsymbol{r}''_\Lambda) - \tilde{\ell}(\boldsymbol{z}', \boldsymbol{H}'_\Lambda, \boldsymbol{r}'_\Lambda) \geq \kappa_1 \left\| \boldsymbol{r}''_\Lambda - \boldsymbol{r}'_\Lambda \right\|_2^2 = \kappa_1 \left(\left\| \boldsymbol{v}'' - \boldsymbol{v}' \right\|_2^2 + \left\| \boldsymbol{e}''_\Lambda - \boldsymbol{e}'_\Lambda \right\|_2^2 \right). \tag{B.21}$$

On the other hand,

$$\begin{aligned} \tilde{\ell}(\boldsymbol{z}', \boldsymbol{H}'_\Lambda, \boldsymbol{r}''_\Lambda) - \tilde{\ell}(\boldsymbol{z}', \boldsymbol{H}'_\Lambda, \boldsymbol{r}'_\Lambda) &= \tilde{\ell}(\boldsymbol{z}', \boldsymbol{H}'_\Lambda, \boldsymbol{r}''_\Lambda) - \tilde{\ell}(\boldsymbol{z}'', \boldsymbol{H}''_\Lambda, \boldsymbol{r}''_\Lambda) + \tilde{\ell}(\boldsymbol{z}'', \boldsymbol{H}''_\Lambda, \boldsymbol{r}''_\Lambda) - \tilde{\ell}(\boldsymbol{z}', \boldsymbol{D}'_\Lambda, \boldsymbol{r}'_\Lambda) \\ &\leq \tilde{\ell}(\boldsymbol{z}', \boldsymbol{H}'_\Lambda, \boldsymbol{r}''_\Lambda) - \tilde{\ell}(\boldsymbol{z}'', \boldsymbol{H}''_\Lambda, \boldsymbol{r}''_\Lambda) + \tilde{\ell}(\boldsymbol{z}'', \boldsymbol{H}''_\Lambda, \boldsymbol{r}'_\Lambda) - \tilde{\ell}(\boldsymbol{z}', \boldsymbol{H}'_\Lambda, \boldsymbol{r}'_\Lambda), \end{aligned} \tag{B.22}$$

where the last inequality holds because $\boldsymbol{r}''$ is the minimizer of $\tilde{\ell}(\boldsymbol{z}'', \boldsymbol{H}'', \boldsymbol{r})$.

We shall prove that $\tilde{\ell}(\boldsymbol{z}', \boldsymbol{H}'_\Lambda, \boldsymbol{r}_\Lambda) - \tilde{\ell}(\boldsymbol{z}'', \boldsymbol{H}''_\Lambda, \boldsymbol{r}_\Lambda)$ is Lipschitz w.r.t. $\boldsymbol{r}$, which implies the Lipschitz of $\boldsymbol{v}'(\boldsymbol{z}, \boldsymbol{D})$ and $\boldsymbol{e}'(\boldsymbol{z}, \boldsymbol{D})$.

$$\begin{aligned} &\nabla_{\boldsymbol{r}} \left(\tilde{\ell}(\boldsymbol{z}', \boldsymbol{H}'_\Lambda, \boldsymbol{r}_\Lambda) - \tilde{\ell}(\boldsymbol{z}'', \boldsymbol{H}''_\Lambda, \boldsymbol{r}_\Lambda) \right) \\ &= \lambda_1 \Big[\boldsymbol{H}'^\top_\Lambda (\boldsymbol{H}'_\Lambda - \boldsymbol{H}''_\Lambda) + (\boldsymbol{H}'_\Lambda - \boldsymbol{H}''_\Lambda)^\top \boldsymbol{H}''_\Lambda + \boldsymbol{H}'^\top_\Lambda (\boldsymbol{z}'' - \boldsymbol{z}') + (\boldsymbol{H}''_\Lambda - \boldsymbol{H}'_\Lambda)^\top \boldsymbol{z}'' \Big]. \end{aligned}$$

Note that $\|\boldsymbol{H}'_\Lambda\|_F$, $\|\boldsymbol{H}''_\Lambda\|_F$ and $\boldsymbol{z}''$ are all uniformly bounded. Hence, there exists uniform constants c_1 and c_2, such that

$$\left\| \nabla_{\boldsymbol{r}} \left(\tilde{\ell}(\boldsymbol{z}', \boldsymbol{H}'_\Lambda, \boldsymbol{r}_\Lambda) - \tilde{\ell}(\boldsymbol{z}'', \boldsymbol{H}''_\Lambda, \boldsymbol{r}_\Lambda) \right) \right\|_2 \leq c_1 \left\| \boldsymbol{H}'_\Lambda - \boldsymbol{H}''_\Lambda \right\|_F + c_2 \left\| \boldsymbol{z}' - \boldsymbol{z}'' \right\|_2,$$

which implies that $\tilde{\ell}(\boldsymbol{z}', \boldsymbol{H}'_\Lambda, \boldsymbol{r}_\Lambda) - \tilde{\ell}(\boldsymbol{z}'', \boldsymbol{H}''_\Lambda, \boldsymbol{r}_\Lambda)$ is Lipschitz with Lipschitz constant $c(\boldsymbol{H}'_\Lambda, \boldsymbol{H}''_\Lambda, \boldsymbol{z}', \boldsymbol{z}'') = c_1 \|\boldsymbol{H}'_\Lambda - \boldsymbol{H}''_\Lambda\|_F + c_2 \|\boldsymbol{z}' - \boldsymbol{z}''\|_2$. Combining this fact with (B.21) and (B.22), we obtain

$$\kappa_1 \left\|\boldsymbol{r}''_\Lambda - \boldsymbol{r}'_\Lambda\right\|_2^2 \leq c(\boldsymbol{H}'_\Lambda, \boldsymbol{H}''_\Lambda, \boldsymbol{z}', \boldsymbol{z}'') \left\|\boldsymbol{r}''_\Lambda - \boldsymbol{r}'_\Lambda\right\|_2.$$

Therefore, $\boldsymbol{r}(\boldsymbol{z}, \boldsymbol{D})$ is Lipschitz and so are $\boldsymbol{v}(\boldsymbol{z}, \boldsymbol{D})$ and $\boldsymbol{e}(\boldsymbol{z}, \boldsymbol{D})$. Note that according to Proposition 9,

$$\begin{aligned}
&\nabla f(\boldsymbol{D}') - \nabla f(\boldsymbol{D}'') \\
=& \mathbb{E}_{\boldsymbol{z}}\left[(\boldsymbol{H}'\boldsymbol{r}' - \boldsymbol{z})\boldsymbol{v}'^\top - (\boldsymbol{H}''\boldsymbol{r}'' - \boldsymbol{z})\boldsymbol{v}''^\top\right] \\
=& \mathbb{E}_{\boldsymbol{z}}\left[\boldsymbol{H}'\boldsymbol{r}'(\boldsymbol{v}' - \boldsymbol{v}'')^\top + (\boldsymbol{H}' - \boldsymbol{H}'')\boldsymbol{r}'\boldsymbol{v}''^\top + \boldsymbol{H}''(\boldsymbol{r}' - \boldsymbol{r}'')\boldsymbol{v}''^\top + \boldsymbol{z}(\boldsymbol{v}'' - \boldsymbol{v}')^\top\right].
\end{aligned}$$

Thus,

$$\begin{aligned}
\left\|\nabla f(\boldsymbol{D}') - \nabla f(\boldsymbol{D}'')\right\|_F &\overset{\zeta_1}{\leq} \mathbb{E}_{\boldsymbol{z}}\Big[\left\|H'\boldsymbol{r}'\right\|_2 \left\|\boldsymbol{v}' - \boldsymbol{v}''\right\|_2 + \left\|\boldsymbol{H}' - \boldsymbol{H}''\right\|_F \left\|\boldsymbol{r}'\boldsymbol{v}''^\top\right\|_F \\
&\qquad + \left\|\boldsymbol{H}''\right\|_F \left\|\boldsymbol{r}' - \boldsymbol{r}''\right\|_2 \left\|\boldsymbol{v}''\right\|_2 + \|\boldsymbol{z}\|_2 \left\|\boldsymbol{v}' - \boldsymbol{v}''\right\|_2\Big] \\
&\overset{\zeta_2}{\leq} \mathbb{E}_{\boldsymbol{z}}\left[(\gamma_1 + \gamma_2 \|\boldsymbol{z}\|_2) \left\|\boldsymbol{H}' - \boldsymbol{H}''\right\|_F\right] \\
&\overset{\zeta_3}{\leq} \gamma_0 \left\|\boldsymbol{D}' - \boldsymbol{D}''\right\|_F,
\end{aligned}$$

where γ_0, γ_1 and γ_2 are all uniform constants. Here, ζ_1 holds because for any function $s(\boldsymbol{z})$, we have $\|\mathbb{E}_{\boldsymbol{z}}[s(\boldsymbol{z})]\|_F \leq \mathbb{E}_{\boldsymbol{z}}[\|s(\boldsymbol{z})\|_F]$. ζ_2 is derived by using the result that $\boldsymbol{r}(\boldsymbol{z}, \boldsymbol{H})$ and $\boldsymbol{v}(\boldsymbol{z}, \boldsymbol{H})$ are both Lipschitz and $\boldsymbol{H}'$, $\boldsymbol{H}''$, $\boldsymbol{r}'$, $\boldsymbol{r}''$, $\boldsymbol{v}'$ and $\boldsymbol{v}''$ are all uniformly bounded. ζ_3 holds because $\boldsymbol{z}$ is uniformly bounded and actually $\|\boldsymbol{H}' - \boldsymbol{H}''\|_F = \|\boldsymbol{D}' - \boldsymbol{D}''\|_F$. Thus, we complete the proof. □

B.7 Proof of stationary point

Theorem 17 (Convergence of $\boldsymbol{D}_t$)**.** *Let $\{\boldsymbol{D}_t\}$ be the optimal basis produced by Algorithm 1 and let $f(\boldsymbol{D})$ be the expected loss function defined in (2.20). Then $\boldsymbol{D}_t$ converges to a stationary point of $f(\boldsymbol{D})$ when t goes to infinity.*

Proof. Since $\frac{1}{t}\boldsymbol{A}_t$ and $\frac{1}{t}\boldsymbol{B}_t$ are uniformly bounded (Proposition 7), there exist sub-sequences of $\{\frac{1}{t}\boldsymbol{A}_t\}$ and $\{\frac{1}{t}\boldsymbol{B}_t\}$ that converge to $\boldsymbol{A}_\infty$ and $\boldsymbol{B}_\infty$ respectively. Then $\boldsymbol{D}_t$ will converge to $\boldsymbol{D}_\infty$. Let $\boldsymbol{W}$ be an arbitrary matrix in $\mathbb{R}^{p\times d}$ and $\{h_k\}$ be any positive sequence that converges to zero.

As g_t is a surrogate function of f_t, for all t and k, we have

$$g_t(\boldsymbol{D}_t + h_k\boldsymbol{W}) \geq f_t(\boldsymbol{D}_t + h_k\boldsymbol{W}).$$

Let t tend to infinity, and note that $f(\boldsymbol{D}) = \lim_{t\to\infty} f_t(\boldsymbol{D})$, we have

$$g_\infty(\boldsymbol{D}_\infty + h_k\boldsymbol{W}) \geq f(\boldsymbol{D}_\infty + h_k\boldsymbol{W}).$$

Note that the Lipschitz of ∇f indicates that the second derivative of $f(\boldsymbol{D})$ is uniformly bounded. By a simple calculation, we can also show that it also holds for $g_t(\boldsymbol{D})$. This fact implies that we can take the first order Taylor expansion for both $g_t(\boldsymbol{D})$ and $f(\boldsymbol{D})$ even when t tends to infinity (because the second order derivatives of them always exist). That is,

$$\mathrm{Tr}(h_k\boldsymbol{W}^\top\nabla g_\infty(\boldsymbol{D}_\infty)) + o(h_k\boldsymbol{W}) \geq \mathrm{Tr}(h_k\boldsymbol{W}^\top\nabla f(\boldsymbol{D}_\infty)) + o(h_k\boldsymbol{W}).$$

By multiplying $\frac{1}{h_k\|\boldsymbol{W}\|_F}$ on both sides and note that $\{h_k\}$ is a positive sequence, it follows that

$$\mathrm{Tr}\left(\frac{1}{\|\boldsymbol{W}\|_F}\boldsymbol{W}^\top\nabla g_\infty(\boldsymbol{D}_\infty)\right)+\frac{o(h_k\boldsymbol{W})}{h_k\|\boldsymbol{W}\|_F}\geq \mathrm{Tr}\left(\frac{1}{\|\boldsymbol{W}\|_F}\boldsymbol{W}^\top\nabla f(\boldsymbol{D}_\infty)\right)+\frac{o(h_k\boldsymbol{W})}{h_k\|\boldsymbol{W}\|_F}.$$

Now let k go to infinity,

$$\mathrm{Tr}\left(\frac{1}{\|\boldsymbol{W}\|_F}\boldsymbol{W}^\top\nabla g_\infty(\boldsymbol{D}_\infty)\right)\geq \mathrm{Tr}\left(\frac{1}{\|\boldsymbol{W}\|_F}\boldsymbol{W}^\top\nabla f(\boldsymbol{D}_\infty)\right).$$

Note that this inequality holds for any matrix $\boldsymbol{W}\in\mathbb{R}^{p\times d}$, so we actually have

$$\nabla g_\infty(\boldsymbol{D}_\infty)=\nabla f(\boldsymbol{D}_\infty).$$

As $\boldsymbol{D}_\infty$ is the minimizer of $g_\infty(\boldsymbol{D})$, we have

$$\nabla f(\boldsymbol{D}_\infty)=\nabla g_\infty(\boldsymbol{D}_\infty)=0.$$

□

References

[AKKS12] Haim Avron, Satyen Kale, Shiva Prasad Kasiviswanathan, and Vikas Sindhwani. Efficient and practical stochastic subgradient descent for nuclear norm regularization. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[Ber99] Dimitri P. Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.

[BM05] Samuel Burer and Renato D. C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005.

[Bot98] Léon Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9), 1998.

[BS98] J. Frédéric Bonnans and Alexander Shapiro. Optimization problems with perturbations: A guided tour. *SIAM Review*, 40(2):228–264, 1998.

[CCS10] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[CLMW11] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11, 2011.

[CR09] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[DBL14] Aaron Defazio, Francis R. Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Neural Information Processing Systems*, pages 1646–1654, 2014.

[Don95] David L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Information Theory*, 41(3):613–627, 1995.

[EBN11] Brian Eriksson, Laura Balzano, and Robert D. Nowak. High-rank matrix completion and subspace clustering with missing data. *CoRR*, abs/1112.5629, 2011.

[EV09] Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2790–2797, 2009.

[FHB01] Maryam Fazel, Haitham Hindi, and Stephen P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American Control Conference*, volume 6, pages 4734–4739. IEEE, 2001.

[FXY13] Jiashi Feng, Huan Xu, and Shuicheng Yan. Online robust PCA via stochastic optimization. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pages 404–412, 2013.

[GQV14] Han Guo, Chenlu Qiu, and Namrata Vaswani. An online algorithm for separating sparse and low-dimensional signal sequences from their sum. *IEEE Trans. Signal Processing*, 62(16):4284–4297, 2014.

[HO14] Cho-Jui Hsieh and Peder A. Olsen. Nuclear norm minimization via active subspace selection. In *Proceedings of the 31st International Conference on Machine Learning*, pages 575–583, 2014.

[JS10] Martin Jaggi and Marek Sulovský. A simple algorithm for nuclear norm regularized problems. In *Proceedings of the 27th International Conference on Machine Learning*, pages 471–478, 2010.

[LCM10] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[LL14] Guangcan Liu and Ping Li. Recovery of coherent data via low-rank dictionary pursuit. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems*, pages 1206–1214, 2014.

[LLY$^+$13] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

[MBPS10] Julien Mairal, Francis R. Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

[MMG15] Morteza Mardani, Gonzalo Mateos, and Georgios B. Giannakis. Subspace learning and imputation for streaming big data matrices and tensors. *IEEE Trans. Signal Processing*, 63(10):2663–2677, 2015.

[NJW01] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of the 15th Annual Conference on Neural Information Processing Systems*, pages 849–856, 2001.

[QVLH14] Chenlu Qiu, Namrata Vaswani, Brian Lois, and Leslie Hogben. Recursive robust PCA or recursive sparse recovery in large but structured noise. *IEEE Trans. Information Theory*, 60(8):5007–5039, 2014.

[RFP10] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[RT14] Peter Richtárik and Martin Takác. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

[SC12] Mahdi Soltanolkotabi and Emmanuel J. Candès. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40:2195–2238, 2012.

[SEC14] Mahdi Soltanolkotabi, Ehsan Elhamifar, and Emmanuel J. Candès. Robust subspace clustering. *The Annals of Statistics*, 42(2):669–699, 2014.

[SL16] Jie Shen and Ping Li. Learning structured low-rank representation via matrix factorization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 500–509, 2016.

[vdV00] A.W. van der Vaart. *Asymptotic statistics*. Cambridge University Press, 2000.

[Vid10] René Vidal. A tutorial on subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2010.

[WXL13] Yu-Xiang Wang, Huan Xu, and Chenlei Leng. Provable subspace clustering: When LRR meets SSC. In *Proceedings of 27th Annual Conference on Neural Information Processing Systems*, pages 64–72, 2013.

[XCM10] Huan Xu, Constantine Caramanis, and Shie Mannor. Principal component analysis with contaminated data: The high dimensional case. In *Proceedings of the 23rd Conference on Learning Theory*, pages 490–502, 2010.

[XZ14] Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.